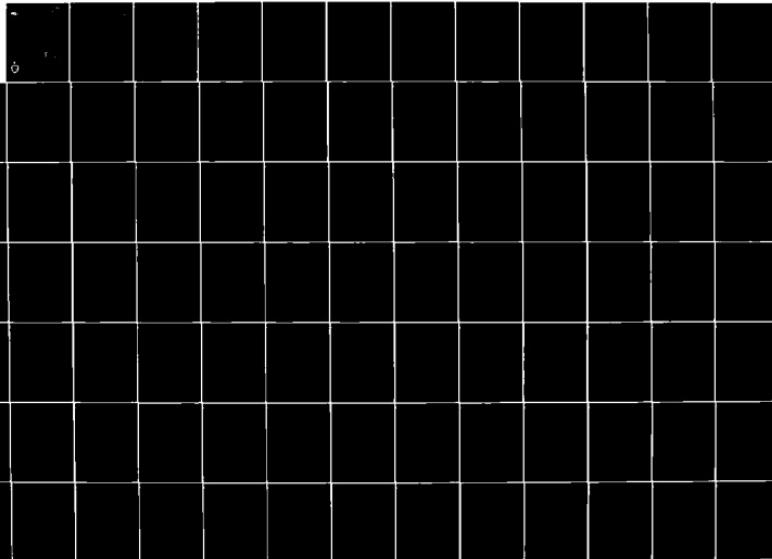


AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

1 > 6
M
M 1 2 3 4 5





-TR-0107

AD [redacted]
Reports Control Symbol
OSD - 1366

2

SUPPLEMENT TO EOSAEL 80
VOLUME II
USER'S MANUAL

PROGRAM LISTINGS FOR EOSAEL 80-B
AND ANCILLARY CODES AGAUS AND FLASH

FEBRUARY 1982

By

R. G. Steinhoff

FILE COPY

DTIC
ELECTED
MAY 14 1982
S D

Approved for public release; distribution unlimited.



US Army Electronics Research and Development Command
Atmospheric Sciences Laboratory
White Sands Missile Range, NM 88002

82 05 11 016

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government endorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM																				
1. REPORT NUMBER ASL-TR-0107	2. GOVT ACCESSION NO. AD-A 114417	3. RECIPIENT'S CATALOG NUMBER																				
4. TITLE (and Subtitle) SUPPLEMENT TO EOSAEL 80 Vol II USER'S MANUAL PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND FLASH	5. TYPE OF REPORT & PERIOD COVERED Final Report																					
7. AUTHOR(s) R. G. Steinhoff	6. PERFORMING ORG. REPORT NUMBER																					
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Atmospheric Sciences Laboratory White Sands Missile Range, NM	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Task AH71612111 H710011																					
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Electronics Research and Development Command Adelphi, MD 20783	12. REPORT DATE February 1982																					
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	13. NUMBER OF PAGES 510																					
	15. SECURITY CLASS. (of this report) UNCLASSIFIED																					
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE <table border="1"><tr><td colspan="2">Accession For</td></tr><tr><td>NTIS GRANT</td><td><input checked="" type="checkbox"/></td></tr><tr><td>DTIC TAB</td><td><input type="checkbox"/></td></tr><tr><td>Unannounced</td><td><input type="checkbox"/></td></tr><tr><td colspan="2">Justification _____</td></tr><tr><td colspan="2">By _____</td></tr><tr><td colspan="2">Distribution _____</td></tr><tr><td colspan="2">Availability _____</td></tr><tr><td>Distr</td><td>Avail. for Spec. 1</td></tr><tr><td>A</td><td></td></tr></table>		Accession For		NTIS GRANT	<input checked="" type="checkbox"/>	DTIC TAB	<input type="checkbox"/>	Unannounced	<input type="checkbox"/>	Justification _____		By _____		Distribution _____		Availability _____		Distr	Avail. for Spec. 1	A	
Accession For																						
NTIS GRANT	<input checked="" type="checkbox"/>																					
DTIC TAB	<input type="checkbox"/>																					
Unannounced	<input type="checkbox"/>																					
Justification _____																						
By _____																						
Distribution _____																						
Availability _____																						
Distr	Avail. for Spec. 1																					
A																						
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)																						
18. SUPPLEMENTARY NOTES																						
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) EOSAEL 80 Computer Codes Transmission Transport and diffusion																						
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Computer listing of the Electro-Optical Systems Atmospheric Effects Library, version 80-B.																						

CONTENTS

INTRODUCTION	7
PROGRAM EOSAEL	9
SUBROUTINE RESET	18
SUBROUTINE COMPLT	19
SUBROUTINE ILLUM	20
FUNCTION SOLARS	21
FUNCTION SMOON	22
FUNCTION JPASCT	23
SUBROUTINE PFUNC	24
SUBROUTINE XSCALE	29
SUBROUTINE SLANT	33
SUBROUTINE TURB	35
FUNCTION DESUB	43
FUNCTION FALPH	44
SUBROUTINE FFT4	45
FUNCTION GAUSS	47
SUBROUTINE MEANYR	48
SUBROUTINE SPECT	49
SUBROUTINE SPREAD	50
SUBROUTINE THETO	51
SUBROUTINE BASCAT	52
SUBROUTINE PFUNC	54
SUBROUTINE BKWD	66
SUBROUTINE CONV	68
SUBROUTINE ELM	69
SUBROUTINE FIND	70
SUBROUTINE FWRD	71
SUBROUTINE GAS	72
SUBROUTINE GMAX	73
SUBROUTINE MATRX	74
SUBROUTINE ROTAT	75
SUBROUTINE SMOOZ	76
SUBROUTINE START	77
SUBROUTINE TRAVRS	79
SUBROUTINE USCA	81
SUBROUTINE SMOKE	82
FUNCTION JPASCT	83
SUBROUTINE CLSMOK	93
SUBROUTINE SCONST	96
SUBROUTINE SMASSP	98
SUBROUTINE STRANS	99
SUBROUTINE WGGEOM	100
FUNCTION QROOT	102
SUBROUTINE XYZINT	103
SUBROUTINE BRATE	108
SUBROUTINE GPUFF	109
SUBROUTINE LZTRAN	111
SUBROUTINE LZIDNM	115
SUBROUTINE DRTRAN	117
SUBROUTINE AMOUNT	130
SUBROUTINE ATMCAL	131
SUBROUTINE AVRG	135
SUBROUTINE CLIMB	136
SUBROUTINE CLDIM	138
SUBROUTINE COMPCL	140
SUBROUTINE CONLEN	143
SUBROUTINE CONVRT	145
FUNCTION CSPHER	147
FUNCTION CWAKE	149
FUNCTION CWIND	151

SUBROUTINE	DIFEQ	154
FUNCTION	DIFFUS	156
FUNCTION	DOTPRD	157
FUNCTION	DTERPI	158
SUBROUTINE	DTERPS	160
SUBROUTINE	DUSTCL	161
FUNCTION	ERF	164
SUBROUTINE	FIT	165
FUNCTION	FUNCT	166
SUBROUTINE	GAMMA	167
FUNCTION	GFUN	168
SUBROUTINE	GRAD2	169
SUBROUTINE	GRAND	170
SUBROUTINE	GREEN	172
SUBROUTINE	GREEN1	173
SUBROUTINE	MOMENT	175
SUBROUTINE	PATH	178
SUBROUTINE	PERP	179
SUBROUTINE	PRECL	180
SUBROUTINE	PRETRN	183
SUBROUTINE	PREVEH	185
SUBROUTINE	RISE	188
SUBROUTINE	RKM	190
SUBROUTINE	SETUP	192
SUBROUTINE	SOURCE	194
SUBROUTINE	TEMP	197
FUNCTION	TMPCAL	198
SUBROUTINE	TRAP	200
SUBROUTINE	TRNCAL	203
SUBROUTINE	TRNCHK	207
SUBROUTINE	TRNCLD	208
SUBROUTINE	UNIT	210
SUBROUTINE	VEHCL	211
SUBROUTINE	VEHTRN	213
SUBROUTINE	VSRC	215
SUBROUTINE	VSUM	217
SUBROUTINE	WIN	218
FUNCTION	WNDCAL	219
SUBROUTINE	NMMW	221
FUNCTION	AB	224
FUNCTION	DOP	225
SUBROUTINE	INTRP	226
SUBROUTINE	MMH20	227
SUBROUTINE	MMIDX	228
SUBROUTINE	MMOXY	229
SUBROUTINE	MMRAN	230
SUBROUTINE	MHSNO	232
SUBROUTINE	MMWFG	234
SUBROUTINE	MMWGS	235
FUNCTION	PFR	237
FUNCTION	PSAT	238
FUNCTION	SUPK	239
SUBROUTINE	CLTRAN	240
SUBROUTINE	LAYRXY	244
SUBROUTINE	CYLXY	245
SUBROUTINE	CLEXTH	247
SUBROUTINE	LISOUT	248
SUBROUTINE	DEFSET	250
SUBROUTINE	CLREAD	251
SUBROUTINE	SCREEN	253
SUBROUTINE	XSCALE	259
SUBROUTINE	SLANT	33
FUNCTION	JPASCT	23
SUBROUTINE	PFUNC	24
SUBROUTINE	CWIC	254
SUBROUTINE	CWIC1	261
SUBROUTINE	CWIC3	264

SUBROUTINE	CWIC4	265
SUBROUTINE	ITAM	267
SUBROUTINE	CINV	274
SUBROUTINE	INTAL	279
SUBROUTINE	CASE1	282
SUBROUTINE	CASE2	283
SUBROUTINE	CASE3	284
SUBROUTINE	CYCLE	285
SUBROUTINE	TREQ	286
SUBROUTINE	FCLOUD	287
SUBROUTINE	ILLUM	290
FUNCTION	SOLARS	291
FUNCTION	SMOON	292
SUBROUTINE	THRMCL	293
SUBROUTINE	SSCLD	294
SUBROUTINE	PFNN	295
SUBROUTINE	MSCLD	297
FUNCTION	ETAIN	298
SUBROUTINE	OVRCST	299
SUBROUTINE	ILLUM	300
FUNCTION	SOLARS	301
FUNCTION	SMOON	302
FUNCTION	G2	303
FUNCTION	E1	304
SUBROUTINE	GRNADE	305
SUBROUTINE	CONCH	309
SUBROUTINE	GOGET	311
SUBROUTINE	PARMS	312
SUBROUTINE	EXTIN	313
SUBROUTINE	UMEAN	314
SUBROUTINE	SUMA	315
SUBROUTINE	LCCAT	316
SUBROUTINE	DATRD	317
SUBROUTINE	LT4M	321
SUBROUTINE	XSCALE	329
SUBROUTINE	SLANT	333
SUBROUTINE	ABSORB	328
SUBROUTINE	CKER	336
SUBROUTINE	FREQSL	337
SUBROUTINE	H20VAP	339
SUBROUTINE	H20410	340
SUBROUTINE	LTPATH	341
SUBROUTINE	MOLSCT	343
SUBROUTINE	NH3	344
SUBROUTINE	NITRIC	345
SUBROUTINE	NITRO	346
SUBROUTINE	NO2	347
SUBROUTINE	OZONE	348
SUBROUTINE	POINT	349
FUNCTION	RESFN	350
SUBROUTINE	SO2	351
SUBROUTINE	UNIMIX	352
SUBROUTINE	UV0ZNE	353
SUBROUTINE	SPOT	354
SUBROUTINE	LT4M	321
SUBROUTINE	XSCALE	329
SUBROUTINE	SLANT	333
SUBROUTINE	PFUNC	324
FUNCTION	SOLARS	321
FUNCTION	SMOON	322
FUNCTION	ALBEDO	361
FUNCTION	BLACK	362
SUBROUTINE	COEFS	363
SUBROUTINE	DIAG	364
SUBROUTINE	INDAT	366

SUBROUTINE	OUTPUT	369
SUBROUTINE	PATHRD	372
SUBROUTINE	ZERO	374
SUBROUTINE	CLIMAT	375
PROGRAM	AGAUS	379
SUBROUTINE	AGXP1	385
SUBROUTINE	AGXP2	390
SUBROUTINE	AGXP3	396
SUBROUTINE	ANGLE	398
FUNCTION	GAMMA	399
SUBROUTINE	GUSET	400
SUBROUTINE	MIEGX	401
SUBROUTINE	WATER	405
SUBROUTINE	GAUS	408
SUBROUTINE	DIMER	409
	BLOCK DATA	410
PROGRAM	FLASH1	411
SUBROUTINE	FLASH	412
SUBROUTINE	DATRD	414
SUBROUTINE	DATWT	416
SUBROUTINE	GETIM	418
SUBROUTINE	IRBLIC	421
SUBROUTINE	VSBLC	422
EOSAEL TEST DATA (NEWRUN)		423
EOSAEL OUTPUT (EOOUT)		426

INTRODUCTION

This listing of EOSAEL 80-B is a supplement to Volume II¹ and supersedes all previous listings.² The current listing is complete as of 8 February 1982 and has revisions one through five incorporated into it.

EOSAEL 80-B differs from EOSAEL 80 in that modules SPOT, LT4M, NMMW, CLIMAT, BASCAT, and TURB have been extensively revised and, therefore, appear with new sequence numbers. All other modules have their original sequencing, except where revisions have been inserted or deleted.

The programs are listed by module with each module followed by its subroutines. Subroutines that have been listed for prior modules in the listing are not repeated in the source listing. The table of contents lists each module along with all its corresponding subroutines and the page number of each subroutine in the listing. The elements Eomain, COMPLT, and RESET, which are always to be resident, appear only at the beginning of the table of contents and the source listing.

Also included herein is a sample input file, NEWRUN, and an output file, EOOUT, produced by using the aforementioned sample input file.

The supplemental codes AGAUS and FLASH are supplied with EOSAEL 80. FLASH is described in appendix A of volume II of the User's Manual¹ and is further described in the comments of the source listing. Operating instructions for the AGAUS code may be found in the comments of the source listing. Manuals for AGAUS are available upon written request from the US Army Atmospheric Sciences Laboratory, White Sands Missile Range, New Mexico.

¹ Shirkey, R. C., and S. G. O'Brien, EOSAEL 80, Volume II, User's Manual, ASL-TR-0073, US Army Atmospheric Sciences Laboratory, White Sands ~~Missile~~ Range, NM, 1981.

² Steinhoff, R. G., Program Listings for EOSAEL 80 and Ancillary Codes AGAUS and FLASH, ASL-TR-0073 (Supplement), US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

```

PROGRAM EOSAEL
C MAIN PROGRAM FOR EOSAEL 80 EOM00010
REAL L0TRNS,LAZTRN,LZTRN,MMTRAN,MMWTRN,IPNAM,IAL,IALB1,IALB2 EOM00020
LOGICAL ISPOT,N16,LOREAD EOM00030
COMMON /SPOTL0/ISPOT,LOREAD,N16 EOM00040
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK EOM00050
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB, EOM00060
1 WNDVEL,WNDDIR,IPASCT EOM00070
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUEOM00080
COMMON /GEOMET/PTS(15),IGEOSW EOM00090
C IOIN - CARD READER EOM00100
IOOUT - PRINTER EOM00110
IPHFUN - UNIT UPON WHICH PHASE FUNCTION DATA RESIDES EOM00120
LOUNIT - UNIT UPON WHICH LT4M ATM DATA RESIDES EOM00130
NDIRTU - UNIT UPON WHICH DRTRAN DATA RESIDES EOM00140
NCLIMT - UNIT FOR CLIMATOLOGICAL DATA EOM00150
KSTOR - AUXILIARY START/RESTART UNIT FOR BASCAT EOM00160
NPLOTU - OPTIONAL UNIT FOR WRITING RESULTS FOR SUBSEQUENT PLOTTING PURPOSES BY THE USER EOM00170
DIMENSION TRAN(16),RADAK(16),RADG(16),IPROGN(20) EOM00180
DIMENSION IDOPGM(20),IPNAM(40) EOM00190
DIMENSION IAL(12),DAT(10) EOM00200
C FOR UNIVAC EOM00210
DATA IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR/ EOM00220
1 5,6,3,8,7,24,25/ EOM00230
C PROGRAM NAMES EOM00240
DATA IPNAM/4HSPT,4HTURB,4HBASC,4HLT4M,4HXSCA,4HSMOK,4HDRTR, EOM00250
1 4HL2TR,4HNMMW,4HCLTR,4HSCRE,4HFCL0,4HOVRC,4HGRNA,4H*, / EOM00260
2 4H*,4H*,4H*,4H*,4HCLIM,4H ,4H ,4HAT , / EOM00270
3 4H ,4HLE ,4HE ,4HAN ,4HAN ,4H ,4HAN ,4HEN , / EOM00280
4 4HUD ,4HST ,4HDE ,4H ,4H ,4H ,4H ,4H , / EOM00290
5 4HATE /
C CARD MNEMONICS EOM00300
DATA IAL/4HEGRU,4HVVIS ,4HFREQ,4HWAVL,4HWVNU,4HRESF, EOM00310
14HTARG,4HRFCVR,4HDESG,4HDBSV,4HBFCL,4HGO / EOM00320
DATA PI,TORRMB,CDEGK/3.14159265,1.33322,273.16/ EOM00330
DATA PTS/15*0.0/ EOM00340
DATA PTS/15*0.0/ EOM00350
ISTART=0 EOM00360
CLDAMT=0. EOM00370
CLDHYT=0. EOM00380
FOGPRB=0. EOM00390
PI2=PI/2. EOM00400
PIRAD=PI/180. EOM00410
TWOP1=2.0*PI EOM00420
WRITE (IOOUT,1060) EOM00430
***** I/O *****
C*** INPUT TO EOSAEL IS CARD ORDER-INDEPENDENT, WITH EACH INPUT RECORD EOM00440
C*** HAVING A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4. THE ONLY EXCEPTION EOM00450
C*** TO THIS RULE IS THE GO SENTINEL CARD, WHICH MUST BE THE LAST RECORD EOM00460
C*** IN THE INPUT SEQUENCE. ALL RECORDS ARE READ IN UNDER THE EOM00470
C*** FORMAT (2A4,1X,10E7.4). INTEGERS MUST BE INPUT AS REAL EOM00480
C*** NUMBERS IN THIS COMMON FORMAT SCHEME. THEY ARE LATER FIXED TO EOM00490
C*** THE INTEGER TYPE. THE IDENTIFIERS FOR EACH OF THE INPUT EOM00500
C*** RECORDS ARE AS FOLLOWS : EOM00510
C----- EOM00520
C----- EOM00530
C----- EOM00540
C CARD IDENTIFIER : EORUN EOM00550
C VARIABLES READ : NUMRUN EOM00560
C NUMRUN - NUMBER OF TIMES EOSAEL DRIVER IS TO BE CYCLED EOM00570
C DEFAULT IS 1. EOM00580
C----- EOM00590
C CARD IDENTIFIER : VIS EOM00600
C VARIABLES READ : VIS,EXTN55,EXTN EOM00610
C VIS - VISIBILITY AT WAVELENGTH OF 0.55 MICRONS (KM) EOM00620
C EXTN55 - EXTINCTION COEFFICIENT AT 0.55 MICRONS (KM**-1) EOM00630
C EXTN - EXTINCTION COEFFICIENT AT INPUT WAVELENGTH (KM**-1) EOM00640
C ** NOTE : IF THE VIS CARD IS NOT INPUT, A WARNING IS PRINTED EOM00650
C AND THE VISIBILITY IS SET TO A DEFAULT VALUE OF 10 KM. EOM00660
C----- EOM00670

```

C ** NOTE : IF EXTN55 IS INPUT AS A VALUE LESS THAN 0.0001, IT EOM00680
 IS SET EQUAL TO THE QUOTIENT 3.912/VIS; IF VIS IS EOM00690
 INPUT AS A VALUE LESS THAN 0.0001, IT IS SET EQUAL TO EOM00700
 THE QUOTIENT 3.912/EXTN55. EOM00710
 ** NOTE : EXTN IS NEEDED ONLY FOR BASCAT EOM00720
 ----- EOM00730
 EOM00740
 EOM00750
 EOM00760
 EOM00770
 EOM00780
 EOM00790
 EOM00800
 EOM00810
 EOM00820
 EOM00830
 EOM00840
 EOM00850
 EOM00860
 EOM00870
 EOM00880
 EOM00890
 EOM00900
 EOM00910
 EOM00920
 EOM00930
 EOM00940
 EOM00950
 EOM00960
 EOM00970
 EOM00980
 EOM00990
 EOM01000
 EOM01010
 EOM01020
 EOM01030
 EOM01040
 EOM01050
 EOM01060
 EOM01070
 EOM01080
 EOM01090
 EOM01100
 EOM01110
 EOM01120
 EOM01130
 EOM01140
 EOM01150
 EOM01160
 EOM01170
 EOM01180
 EOM01190
 EOM01200
 EOM01210
 EOM01220
 EOM01230
 EOM01240
 EOM01250
 EOM01260
 EOM01270
 EOM01280
 EOM01290
 EOM01300
 EOM01310
 EOM01320
 EOM01330
 EOM01340
 EOM01350
 EOM01360
 EOM01370

C** ONLY ONE OF THE FOLLOWING THREE CARDS MAY BE INPUT FOR A GIVEN
 C** CYCLE OF EOSAEL. IF NONE OF THESE CARDS IS PRESENT, AN ERROR
 C** MESSAGE IS PRINTED AND EXECUTION IS TERMINATED.

CARD IDENTIFIER : FREQ
 VARIABLES READ : FREQ1, FREQ2, MULDV
 FREQ1 - LOWER INPUT FREQUENCY (GHZ)
 FREQ2 - HIGHER INPUT FREQUENCY (GHZ)
 MULDV - FREQUENCY INCREMENT FOR SPOT AND/OR LT4M (GHZ)

CARD IDENTIFIER : WAVL
 VARIABLES READ : WAVE1, WAVE2, MULDV
 WAVE1 - SHORTER INPUT WAVELENGTH (MICRONS)
 WAVE2 - LONGER INPUT WAVELENGTH (MICRONS)
 MULDV - WAVELENGTH INCREMENT FOR SPOT AND/OR LT4M (MICRONS)

CARD IDENTIFIER : WVNUM
 VARIABLES READ : WVNUM1, WVNUM2, MULDV
 WVNUM1 - LOWER INPUT WAVENUMBER (CM**-1)
 WVNUM2 - HIGHER INPUT WAVENUMBER (CM**-1)
 MULDV - WAVENUMBER INCREMENT FOR SPOT AND/OR LT4M (CM**-1)

C** THE NEXT CARD DETERMINES WHETHER A SENSOR RESPONSE FUNCTION
 C** IS DESIRED FOR BROAD BAND CALCULATIONS. THIS OPTION IS
 C** INVOKED ONLY IF THIS CARD IS PRESENT.

CARD IDENTIFIER : RESF
 VARIABLES READ : NONE HERE - SEE SPOT OR LT4M WRITEUP FOR
 PROPER PLACEMENT OF RESPONSE FUN CARDS.

C** THE NEXT FIVE CARDS COMprise THE GEOMETRICAL OPTION OF EOSAEL.
 C** THIS OPTION IS USEFUL FOR EOSAEL RUNS WHERE SEVERAL MODULES
 C** EXAMINE DIFFERENT ATMOSPHERIC OBSCURATION EFFECTS ALONG THE
 C** SAME PHYSICAL PATH. THE GEOMETRICAL OPTION ASSURES THAT THE
 C** POINTS OF REFERENCE IN THE SCENARIO UNDER EXAMINATION ARE
 C** CONSISTENTLY SPECIFIED FOR ALL MODULES. IT SHOULD BE NOTED
 C** THAT THIS OPTION IS ACTIVATED WHENEVER ANY OF THE FIVE CARDS IS
 C** ENCOUNTERED. ONCE THE OPTION IS ACTIVATED IT IS IMPORTANT THAT
 C** AT LEAST THE FIRST TWO CARDS (TARG AND RCVR) BE INPUT TO
 C** DEFINE THE PHYSICAL PATH, SINCE THIS OPTION WILL OVERRIDE
 C** POSITIONS OR LENGTHS CONTAINED IN NORMAL INPUT TO ALL MODULES.
 C** THE GEOMETRICAL INPUT CONSISTS OF FIVE SETS OF COORDINATES
 C** WHICH OBEY THE FOLLOWING CONVENTIONS :
 C**
 C** (A) ALL COORDINATES ARE DIMENSIONED IN KILOMETERS
 C** (B) THE Z-AXIS IS POSITIVE UPWARD
 C** (C) THE Y-AXIS POINTS NORTH
 C** (D) THE X-AXIS POINTS EAST

C** THE FIVE GEOMETRICAL CARDS ARE AS FOLLOWS :

CARD IDENTIFIER : TARG
 VARIABLES READ : PTS(1), PTS(2), PTS(3)
 PTS(1-3) - COORDINATES OF THE TARGET (FOR THE DRTRAN MODULE,
 THESE ARE THE COORDINATES OF THE TRANSMITTER).

CARD IDENTIFIER : RCVR
 VARIABLES READ : PTS(4), PTS(5), PTS(6)
 PTS(4-6) - COORDINATES OF THE RECEIVER OR SEEKER

CARD IDENTIFIER : DESG
 VARIABLES READ : PTS(7),PTS(8),PTS(9)
 PTS(7-9) - COORDINATES OF THE DESIGNATOR OR SOURCE EOM01380
 EOM01390
 EOM01400
 EOM01410
 EOM01420
 EOM01430
 EOM01440
 EOM01450
 EOM01460
 EOM01470
 EOM01480
 EOM01490
 EOM01500
 EOM01510
 EOM01520
 EOM01530
 EOM01540
 EOM01550
 EOM01560
 EOM01570
 EOM01580
 EOM01590
 EOM01600
 EOM01610
 EOM01620
 EOM01630

CARD IDENTIFIER : OBSV
 VARIABLES READ : PTS(10),PTS(11),PTS(12)
 PTS(10-12) - COORDINATES OF THE OBSERVER USED BY DRTRAN EOM01420
 EOM01430
 EOM01440
 EOM01450
 EOM01460
 EOM01470
 EOM01480
 EOM01490
 EOM01500
 EOM01510
 EOM01520
 EOM01530
 EOM01540
 EOM01550
 EOM01560
 EOM01570
 EOM01580
 EOM01590
 EOM01600
 EOM01610
 EOM01620
 EOM01630

CARD IDENTIFIER : BFCL
 VARIABLES READ : PTS(13),PTS(14),PTS(15)
 PTS(13-15) - COORDINATES OF THE CENTER OF THE CLOUD ELLIPSOID
 USED BY BASCAT AND FCLOUD EOM01420
 EOM01430
 EOM01440
 EOM01450
 EOM01460
 EOM01470
 EOM01480
 EOM01490
 EOM01500
 EOM01510
 EOM01520
 EOM01530
 EOM01540
 EOM01550
 EOM01560
 EOM01570
 EOM01580
 EOM01590
 EOM01600
 EOM01610
 EOM01620
 EOM01630

** THE NEXT INPUT CARD IS THE CLIMATOLOGICAL OPTION CARD. THIS
 ** OPTION ALLOWS USER INPUT OF METEOROLOGICAL PARAMETERS DIRECTLY
 ** OR AUTOMATIC INPUT OF CLIMATOLOGY DATA CHARACTERISTIC OF
 ** WEST GERMANY. IF THIS OPTION IS INVOKED ALL MODULES WILL USE
 ** THIS DATA, I.E. MET DATA THAT HAS BEEN INPUT TO A SPECIFIC
 ** MODULE WILL BE OVERRIDDEN.

CARD IDENTIFIER : CLIMAT
 VARIABLES READ : ICLMAT,LOCAT,MONTH,NHOUR,IWIND,NPRT
 ** OR ** ICLMAT,IPASCT,TEMP,PRESS,RH,AH,DP,VIS,WNDVEL,
 WINDDIR EOM01520
 EOM01530
 EOM01540
 EOM01550
 EOM01560
 EOM01570
 EOM01580
 EOM01590
 EOM01600
 EOM01610
 EOM01620
 EOM01630

LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER
 (1-4) FOR CENTRAL EUROPE AND
 (5-10) FOR MID-EAST.

L = 1 - EUROPEAN LOWLANDS,
 L = 2 - EUROPEAN RHINE VALLEY,
 L = 3 - EUROPEAN HIGHLANDS,
 L = 4 - EUROPEAN ALPINE,
 L = 5 - MIDEAST DESERTS,
 L = 6 - MIDEAST COASTAL,
 L = 7 - MIDEAST PERSIAN GULF,
 L = 8 - MIDEAST RED SEA,
 L = 9 - MIDEAST EASTERN MOUNTAINS, AND
 L = 10 - MIDEAST INDUS VALLEY.

MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR.
 MONTH IS USED TO SELECT THE SEASON WHICH IS
 APPLICABLE TO THE REGION LOCAT.

NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL
 STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OF
 FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14,
 AND 15-19.

IWIND - *** NOT USED ***

NPRT - A PRINT SELECTOR.
 NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA.
 NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD
 DEVIATIONS, AND PERCENT OCCURRENCES.

ICLMAT = 2.: USER INPUT QUANTITIES EOM01820
 IPASCT = PASQUILL STABILITY CATEGORY VALID RANGE =1.-6.(A-F) EOM01830
 TEMP = TEMPERATURE IN DEGREES C EOM01840
 PRESS = PRESSURE IN MB (SEA LEVEL IF ICLMAT=1) EOM01850
 RH = RELATIVE HUMIDITY IN % EOM01860
 AH = ABSOLUTE HUMIDITY - DEFINED HERE AS THE H2O VAPOR
 DENSITY IN G/M**3. EOM01880
 DP = DEW POINT TEMPERATURE IN DEGREES C EOM01890
 VIS = VISIBILITY IN KM EOM01900
 WNDVEL = WIND VELOCITY IN M/S (DEPNT UPON IWIND IF ICLMAT=1) EOM01910
 WINDDIR = WIND DIRECTION IN DEGREES (IF ICLMAT=1 WILL BE
 MOST PROBABLE DIRECTION) EOM01920
 EOM01930
 EOM01940
 EOM01950
 EOM01960
 EOM01970
 EOM01980

** THE FOLLOWING CARDS ARE ALSO READ IN UNDER THE COMMON FORMAT
 ** USED ABOVE. THE INFORMATION ON EACH OF THESE RECORDS DETERMINES
 ** WHICH MODULES ARE SELECTED AND HOW MANY TIMES THE MODULES ARE

*** TO BE CYCLED WITHIN EACH CYCLE OF THE EOSAEL DRIVER. EOM01990
EOM02000
EOM02010
EOM02020
EOM02030
EOM02040
EOM02050
EOM02060
EOM02070
EOM02080
EOM02090
EOM02100
EOM02110
EOM02120
EOM02130
EOM02140
EOM02150
EOM02160
EOM02170
EOM02180
EOM02190
EOM02200
EOM02210
EOM02220
EOM02230
EOM02240
EOM02250
EOM02260
EOM02270
EOM02280
EOM02290
EOM02300

CARD IDENTIFIER : (SEE MODULE IDENTIFIERS BELOW)
VARIABLES READ : IDOPGM
IDOPGM - NUMBER OF TIMES THE SELECTED MODULE IS TO
BE CYCLED WITHIN EACH EOSAEL CYCLE - DEFAULT IS ONE.

NO.	MODULE IDENTIFIER	RANGES	FREQ(GHZ)
1	SPOT	.25-2.,3.-5.,8.-12.*	
2	TURB	LT 14.	*
3	BASCAT	ANY WAVELENGTH IN DATA FILE IPHFUN	
4	LT4M	.25-2.,3.-5.,8.-12.*	
5	XSCALE	1.06,3-5,8-12.*	
6	SMOKE	4-1.2,3-5,8-12.*	
7	DRTRAN	.4-1.1,3.5-4.,8.5-12.*	94.-140.
8	LZTRAN	.8-11.*	
9	NMMW		10-350
10	CLTRAN	.2-2.,3.-5.,8.-12.*	
11	SCREEN	N/A	
12	FCLOUD	ANY WAVELENGTH IN DATA FILE IPHFUN	
13	OVRCST	ANY WAVELENGTH	
14	GRNADE	SAME AS SMOKE	

*** NOTE : THE DATA SPECIFIC TO EACH MODULE MUST BE INPUT IN
THE SEQUENCE IN THE ABOVE LIST.

CARD IDENTIFIER : GO
VARIABLES READ : NONE
END OF READ SENTINEL (MUST BE LAST CARD READ).

```

***** NUMRUN=1 EOM02310
IRFLAG=0 EOM02320
READ (IOIN,1000) IALB1,IALB2,DAT(L),L=1,10 EOM02330
IF (IALB1.NE.IAL(1)) GO TO 10 EOM02340
NUMRUN=IFIX(DAT(1)) EOM02350
IF (NUMRUN.EQ.0) NUMRUN=1 EOM02360
GO TO 20 EOM02370
C SET FLAG IF EORUN IS NOT THE FIRST CARD EOM02380
10 IRFLAG=1 EOM02400
20 CONTINUE EOM02410
DO 580 JRUN=1,NUMRUN EOM02420
C INITIALIZATION EOM02430
DO 30 I=1,20 EOM02440
C PROGRAM SELECTOR EOM02450
IPROGN(I)=0 EOM02460
C PROGRAM CYCLE DEFAULT EOM02470
30 IDOPGM(I)=1 EOM02480
C TRANSMISSIONS EOM02490
LOTRNS=1. EOM02500
XSTRN=1. EOM02510
SMKTRN=1. EOM02520
DRTRN=1. EOM02530
LZTRN=1. EOM02540
MMWTRN=1. EOM02550
CLTRN=1.0 EOM02560
GRNTRN=1.0 EOM02570
IF (JRUN.GT.1) WRITE (IOOUT,1070) JRUN EOM02580
FREQUENCY, WAVELENGTH, WAVENUMBER INDICATOR EOM02590
IFW=0 EOM02600
C GEOMETRICAL OPTION SWITCH EOM02610
IGEOSW=0 EOM02620
C SENSOR RESPONSE FUNCTION OPTION SWITCH EOM02630
NR=0 EOM02640
VIS=0.0 EOM02650
EXTN55=0.0 EOM02660
EXTN=0.0 EOM02670
DO 220 J=1,25 EOM02680

```

```

C      IF (<IRFLAG.EQ.1) WRITE (IOOUT,1010)
EOM02690
SUPPRESS READ IN CASE FIRST CARD PREVIOUSLY READ WASNT FORUN.
EOM02700
IF (<IRFLAG.EQ.0) READ (IOIN,1000) IALB1,IALB2,(DAT(L),L=1,10)
EOM02710
IRFLAG=0
EOM02720
INOPT=0
EOM02730
IF (<J.EQ.25) GO TO 230
EOM02740
DO 40 KK=1,12
EOM02750
C      CHECK FOR CARD TYPES, NOT PROGRAM SELECTOR
EOM02760
IF (<IALB1.NE.IAL(KK)) GO TO 40
EOM02770
INOPT=KK
EOM02780
IF (<INOPT.GE.3.AND.INOPT.LE.5) IFW=INOPT
EOM02790
C      GO CARD FOUND
EOM02800
IF (<INOPT.EQ.12) GO TO 250
EOM02810
GO TO 80
EOM02820
C      40  CONTINUE
EOM02830
C      SEARCH FOR PROGRAMS HERE
EOM02840
DO 50 KK=1,20
EOM02850
IF (<IALB1.NE.IPNAM(KK)) GO TO 50
EOM02860
IPRGMN(KK)=KK
EOM02870
IF (<DAT(1).GT.1,0) IDOPGM(KK)=IFIX(DAT(1))
EOM02880
IF (KK.EQ.20) GO TO 60
EOM02890
GO TO 220
EOM02900
50      CONTINUE
EOM02910
GO TO 240
EOM02920
C      CLIMATOLOGICAL OPTION INVOKED
EOM02930
60      ICLMAT=IFIX(DAT(1))
EOM02940
IF (<ICLMAT.EQ.2) GO TO 70
EOM02950
LOCAT=IFIX(DAT(2))
EOM02960
MONTH=IFIX(DAT(3))
EOM02970
NHOUR=IFIX(DAT(4))
EOM02980
IWIND=IFIX(DAT(5))
EOM02990
NPRT=IFIX(DAT(6))
EOM03000
GO TO 220
EOM03010
70      IPASCT=IFIX(DAT(2))
EOM03020
TEMP=DAT(3)
EOM03030
PRESS=DAT(4)
EOM03040
RH=DAT(5)
EOM03050
AH=DAT(6)
EOM03060
DP=DAT(7)
EOM03070
VIS=DAT(8)
EOM03080
WNDVEL=DAT(9)
EOM03090
WNDDIR=DAT(10)
EOM03100
GO TO 220
EOM03110
C      GEOMETRICAL OPTION INVOKED
EOM03120
C      80  IF (<INOPT.GT.6) GO TO 90
EOM03130
C      CARD SETUP SWITCHING
EOM03140
GO TO (210,160,170,180,190,200),INOPT
EOM03150
90      LPTSSW=INOPT-6
EOM03160
IGEOSW=1
EOM03170
DO 150 K=1,3
EOM03180
GO TO <100,110,120,130,140>,LPTSSW
EOM03190
100     PTS(K)=DAT(K)
EOM03200
GO TO 150
EOM03210
110     PTS(K+3)=DAT(K)
EOM03220
GO TO 150
EOM03230
120     PTS(K+6)=DAT(K)
EOM03240
GO TO 150
EOM03250
130     PTS(K+9)=DAT(K)
EOM03260
GO TO 150
EOM03270
140     PTS(K+12)=DAT(K)
EOM03280
150     CONTINUE
EOM03290
GO TO 220
EOM03300
C      VISIBILITY CARD
EOM03310
160     VIS=DAT(1)
EOM03320
EXTN55=DAT(2)
EOM03330
EXTN=DAT(3)
EOM03340
GO TO 220
EOM03350
C      FREQUENCY CARD
EOM03360
170     FREQ1=DAT(1)
EOM03370
FREQ2=DAT(2)
EOM03380

```

```

        MULDV=IFIX(DAT(3))          E0M03390
        GO TO 220                  E0M03400
C      WAVELENGTH CARD          E0M03410
180    WAVE1=DAT(1)             E0M03420
        WAVE2=DAT(2)             E0M03430
        MULDV=IFIX(DAT(3))          E0M03440
        GO TO 220                  E0M03450
C      WAVENUMBER CARD          E0M03460
190    WVNUM1=DAT(1)             E0M03470
        WVNUM2=DAT(2)             E0M03480
        MULDV=IFIX(DAT(3))          E0M03490
        GO TO 220                  E0M03500
C      SENSOR OPTION INVOKED   E0M03510
200    NR=1                     E0M03520
        GO TO 220                  E0M03530
210    WRITE (IOOUT,1010)         E0M03540
220    CONTINUE                 E0M03550
C      IF (IFW,NE,0) GO TO 250   E0M03560
C      ERROR CHECK ON WAVENUMBER, WAVELENGTH, OR FREQUENCY
        WRITE (IOOUT,1020)         E0M03580
        GO TO 580                  E0M03590
230    WRITE (IOOUT,1030)         E0M03600
        GO TO 250                  E0M03610
C      UNKNOWN CARD TYPE       E0M03620
240    WRITE (IOOUT,1040) IALB1,IALB2
250    CONTINUE                 E0M03630
C      SELECT FREQUENCY, WAVELENGTH, OR WAVENUMBER
        IF (IFW-4) 260,270,280   E0M03640
260    WVNUM1=FREQ1/30          E0M03660
        WVNUM2=FREQ2/30          E0M03670
        WAVE1=0                   E0M03680
        IF (FREQ2,GT,,0001) WAVE1=3.E+05/FREQ2
        WAVE2=3.E+05/FREQ1        E0M03700
        GO TO 290                  E0M03710
270    WVNUM1=0                   E0M03720
        IF (WAVE2,GT,,0001) WVNUM1=1.E+04/WAVE2
        WVNUM2=1.E+04/WAVE1        E0M03730
        FREQ1=30.*WVNUM1           E0M03740
        FREQ2=30.*WVNUM2           E0M03750
        GO TO 290                  E0M03770
280    FREQ1=30.*WVNUM1           E0M03780
        FREQ2=30.*WVNUM2           E0M03790
        WAVE1=0                   E0M03800
        IF (WVNUM2,GT,,0001) WAVE1=1.E+04/WVNUM2
        WAVE2=1.E+04/WVNUM1        E0M03810
290    CONTINUE                 E0M03820
        IF (VIS,LT,,0001,AND,EXTN55,LT,0001) WRITE (IOOUT,1050)
        IF (VIS,LT,,0001,AND,EXTN55,LT,0001) VIS=10.
        IF (EXTN55,GT,,0001) VIS=3.912/EXTN55
        IF (VIS,GT,,0001) EXTN55=3.912/VIS
C      OUTPUT INFORMATION
        WRITE (IOOUT,1080)           E0M03840
        DO 300 I=1,20              E0M03850
300    IF (IPROGN(I),EQ,I) WRITE (IOOUT,1090) IPNAM(I),IPNAM(I+20)
        WRITE (IOOUT,1100) WVNUM1,WVNUM2,WAVE1,WAVE2,FREQ1,FREQ2
C      CLIMAT USES UNIT NCLIMT
        IF (ICLMAT,EQ,1) CALL CLIMAT(LOCAT,MONTH,NHOUR,IWIND,NPRT,TEMP,
        ,PRESS,RH,AH,DP,VIS,WNDVEL,WNDDIR,IPASCT)          E0M03940
        WRITE (IOOUT,1110) VIS
        IF (ICLMAT,EQ,2) ICLMAT=1
C***** SPOT CONTRAST PGM *****
        IF (IPROGN(1),NE,1) GO TO 320
        IPGM1=IDOPGM(1)
        WRITE (IOOUT,1130)
        DO 310 I=1,IPGM1
C      SPOT USES UNITS: IPHFUN - PHASE FUNCTION; LOUNIT - LT4M DATA
310    CALL SPOT(WVNUM1,WVNUM2,VIS,NR,IERR,MULDV)          E0M04040
        CALL RESET(IERR)
C***** TURBULENCE PGM *****
320    IF (IPROGN(2),NE,2) GO TO 340

```

```

IPGM2=IDOPGM(2)
WRITE (IOOUT,1140)                                     EOM04090
DO 330 I=1,IPGM2                                      EOM04100
330 CALL TURB(WAVE1,IERR)
CALL RESET(IERR)
C***** LASER MULTIPLE SCATTERING PGM *****
340 IF (IPROGN(3).NE.3) GO TO 360                     EOM04110
IPGM3=IDOPGM(3)                                       EOM04120
ISPOT=.FALSE.
WRITE (IOOUT,1150)                                     EOM04130
DO 350 I=1,IPGM3                                      EOM04140
C BASCAT USES UNIT IPHFUN FOR PHASE FUNCTION DATA
350 CALL BASCAT(WAVE1,EXTH,IERR)                      EOM04150
CALL RESET(IERR)
C***** LT4M PGM *****
360 IF (IPROGN(4).NE.4) GO TO 380                     EOM04160
C LT4M READS ATM DATA FROM LOUUNIT
ISPOT=.FALSE.
LREAD=.TRUE.
IPGM4=IDOPGM(4)                                       EOM04170
DO 370 I=1,IPGM4                                      EOM04180
CALL LT4M(H1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN,MODEL,EOM04190
     VIS,WNUM1,WNUM2,T1,ICLMAT,IERR,NR,IHAZE,MULDV) EOM04200
370 LOTRNS=L0TRNS*TAN(1)                                EOM04210
CALL RESET(IERR)
C***** XSCALE EXTINCTION PGM *****
380 IF (IPROGN(5).NE.5) GO TO 400                     EOM04220
WRITE (IOOUT,1160)                                     EOM04230
IPGM5=IDOPGM(5)                                       EOM04240
DO 390 I=1,IPGM5                                      EOM04250
CALL XSCALE(WAVE1,VIS,EXTN55,XTRN,IERR,0,0,0.,0.)   EOM04260
390 XSTRN=XSTRN*XTRN                                 EOM04270
CALL RESET(IERR)
C***** SMOKE PGM *****
400 IF (IPROGN(6).NE.6) GO TO 420                     EOM04280
WRITE (IOOUT,1170)                                     EOM04290
IPGM6=IDOPGM(6)                                       EOM04300
DO 410 N=1,IPGM6                                      EOM04310
CALL SMOKE(WAVE1,ICLMAT,STRANS,IERR)                 EOM04320
410 SMKTRN=SMKTRN*STRANS                            EOM04330
CALL RESET(IERR)
C***** DRTRAN PGM *****
420 IF (IPROGN(7).NE.7) GO TO 440                     EOM04340
WRITE (IOOUT,1180)                                     EOM04350
IPGM7=IDOPGM(7)                                       EOM04360
HOLDWV=WAVE1                                         EOM04370
IF (IFW.EQ.1) WAVE1=WAVE2                            EOM04380
DO 430 N=1,IPGM7                                      EOM04390
C DRTRAN USES HDIRTU FOR DATA
IF (N.GT.1) WRITE (IOOUT,1120)                         EOM04400
CALL DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)                 EOM04410
430 DRTRN=DRTRN*TRNLOS                               EOM04420
WAVE1=HOLDWV                                         EOM04430
CALL RESET(IERR)
C***** LASER TRANSMISSION PGM *****
440 IF (IPROGN(8).NE.8) GO TO 460                     EOM04440
WRITE (IOOUT,1190)                                     EOM04450
IPGM8=IDOPGM(8)                                       EOM04460
DO 450 I=1,IPGM8                                      EOM04470
CALL LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)                EOM04480
450 LZTRN=LZTRN*LAZTRN                             EOM04490
CALL RESET(IERR)
C***** NEAR MILLIMETER WAVE PGM *****
460 IF (IPROGN(9).NE.9) GO TO 480                     EOM04500
WRITE (IOOUT,1200)                                     EOM04510
IPGM9=IDOPGM(9)                                       EOM04520
DO 470 I=1,IPGM9                                      EOM04530
CALL NMWW(FREQ1,ICLMAT,MMTRAN,IERR)                 EOM04540
470 MMWTRN=MMWTRN*MMTRAN                            EOM04550
CALL RESET(IERR)

```

```

***** CLOUD TRANSMISSION PGM *****
480 IF (IPROGN(10).NE.10) GO TO 500 EOM04790
    IPGM10=IDOPGM(10)
        DO 490 I=1,IPGM10 EOM04800
        WRITE (IOOUT,1210) EOM04810
        CALL CLTRAN(CTRANS,WAVE1,I,IERR) EOM04820
    490 CLTRN=CLTRN*CTRANS EOM04830
        CALL RESET(IERR) EOM04840
***** CWIC MUNITION EXPENDITURES/INVERSE STATIC TARGET DETECTION PGM *****
500 IF (IPROGN(11).NE.11) GO TO 520 EOM04850
    WRITE (IOOUT,1220) EOM04860
    IPGM11=IDOPGM(11)
        DO 510 I=1,IPGM11 EOM04870
    510 CALL SCREEN(IERR,ICLMAT) EOM04880
        CALL RESET(IERR) EOM04890
***** FINITE CLOUD RADIATIVE TRANSFER PGM *****
520 IF (IPROGN(12).NE.12) GO TO 540 EOM04950
    WRITE (IOOUT,1230) EOM04960
    IPGM12=IDOPGM(12)
C     FCLOUD USES IPHFUN FOR PHASE FUNCTION DATA EOM04970
        DO 530 I=1,IPGM12 EOM04980
    530 CALL FCLOUD(WAVE1,FTRANS,IERR) EOM04990
        CALL RESET(IERR) EOM05000
***** OVERCAST SKY RADIATIVE TRANSFER PGM *****
540 IF (IPROGN(13).NE.13) GO TO 560 EOM05010
    WRITE (IOOUT,1240) EOM05020
    IPGM13=IDOPGM(13)
        DO 550 I=1,IPGM13 EOM05030
    550 CALL OVRCS(CTRANS,WAVE1,OTRANS,IERR) EOM05040
        CALL RESET(IERR) EOM05050
***** SELF-SCREENING SMOKE GRENADE PGM *****
560 IF (IPROGN(14).NE.14) GO TO 575 EOM05060
    WRITE (IOOUT,1250) EOM05070
    IPGM14=IDOPGM(14)
        DO 570 I=1,IPGM14 EOM05080
    570 CALL GRNAD(E,WAVE1,ICLMAT,GRTRAN,IERR) EOM05090
        GRNTRN=GRNTRN*GRTRAN EOM05100
        CALL RESET(IERR) EOM05110
***** *****
575 IF((IPROGN(4).GT.0.OR.IPROGN(5).GT.0.OR.IPROGN(6).GT.0.OR.IPROGN(7).
+ .GT.0.OR.IPROGN(8).GT.0.OR.IPROGN(9).GT.0.OR.IPROGN(10).GT.0.OR.
+ .IPROGN(14).GT.0) GO TO 576 EOM05120
    GO TO 580 EOM05130
576 CALL COMPLT(LOTRNS,XSTRN,SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN) EOM05140
580 CONTINUE EOM05150
    WRITE (IOOUT,1260) EOM05160
    STOP EOM05170
C
C
1000 FORMAT(2A4,1X,1E7.4) EOM05180
1010 FORMAT(1H0,20X,75H***EOMAIN WARNING*** EORUN CYCLE CARD OUT OF SEQUE EOM05190
    UENCE, DEFAULT TO ONE CYCLE //) EOM05200
1020 FORMAT(1H0,20X,74H***EOMAIN ERROR*** FREQ, WAVL, OR WNUM CARD WAS EOM05270
    1 NOT INPUT, RUN TERMINATED //) EOM05280
1030 FORMAT(1H0,20X,46H***EOMAIN ERROR*** END OF READ SENTINEL ABSENT / EOM05290
    1/,1X,20X,28HRESULTS MAY BE UNPREDICTABLE) EOM05300
1040 FORMAT(1H0,20X,80H***EOMAIN ERROR*** INPUT CARD DETECTED WHICH DOES EOM05310
    IS NOT MATCH CORRECT INPUT FORMAT//,1X,20X,13HTHE CARD WAS: ,2X,2A4) EOM05320
1050 FORMAT(1H0,20X,24H*** EOSAEL WARNING ***,/ ,1X,20X,15HVISIBILITY EOM05330
    1 AND 47HEXTINCT = 0.0, VISIBILITY CHANGED TO 10.0 KM//) EOM05340
1060 FORMAT(1H1,/////,1X,50X,30(1H*),/ ,1X,50X,1H*,28X,1H*,/ EOM05350
    1 ,1X,50X,30H* ELECTRO-OPTICAL SYSTEMS *,/ ,1X,50X,1H*, EOM05360
    2 ,28X,1H*,/ ,1X,50X,30H* ATMOSPHERIC EFFECTS *,/ EOM05370
    3 ,1X,50X,1H*,28X,1H*,/ ,1X,50X,25H* LIBRARY EOM05380
    4 ,5H *,/ ,1X,50X,1H*,28X,1H*,/ ,1X,50X,30(1H*)) EOM05390
1070 FORMAT(1H1,/////,58X,11HRUN NUMBER,12) EOM05400
1080 FORMAT(///1X,51X,28HINDIVIDUAL MODULES SELECTED) EOM05410
1090 FORMAT(1X,62X,2A4) EOM05420
1100 FORMAT(1H0,63X,9HBEGINNING,12X,6HENDING,/,39X,14HWAVENUMBER(CM* EOM05430
    1 4H*-1),6X,F10.3,10X,F10.3,/,39X,19HWAVELENGTH(MICRONS), EOM05440

```

2 5X,F10.3,10X,F10.3,/,39X,14HFREQUENCY(GHZ),5X,F15.3,5X,
3 F15.3,/,/
1110 FORMAT(1H0,62X,10HVISIBILITY,/,62X,F5.2,3H KM) E0M05450
1120 FORMAT(1H1) E0M05460
1130 FORMAT(1H1,40X,20HSPOT CONTRAST MODULE //) E0M05480
1140 FORMAT(1H1,40X,17HTURB LASER MODULE //) E0M05490
1150 FORMAT(1H1,40X,30HBASCAT LASER SCATTERING MODULE //) E0M05500
1160 FORMAT(1H1,40X,46HXSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE
1170 FORMAT(1H1,45X,19HSMOKE MODEL MODULE ///) E0M05510
1180 FORMAT(1H1,40X,26H DIRT TRANSMISSION MODULE ///) E0M05520
1190 FORMAT(1H1,40X,28H LASER TRANSMITTANCE MODULE //) E0M05530
1200 FORMAT(1H1,45X,29H NEAR MILLIMETER WAVE MODULE ///) E0M05540
1210 FORMAT(1H1,40X,27HCLOUD TRANSMITTANCE MODULE ///) E0M05550
1220 FORMAT(1H1,20X,43HCWIC MUNITION EXPENDITURES / INVERSE STATIC
1 24H TARGET DETECTION MODULE) E0M05560
1230 FORMAT(1H1,40X,38HFINITE CLOUD RADIATIVE TRANSFER MODULE //) E0M05570
1240 FORMAT(1H1,40X,38HOVERCAST SKY RADIATIVE TRANSFER MODULE //) E0M05580
1250 FORMAT(1H1,40X,35HSELF-SCREENING SMOKE GRENADE MODULE //) E0M05590
1260 FORMAT(1X,/,/,1X,50X,14HENDE03AE RUN)
END E0M05600
 E0M05610
 E0M05620
 E0M05630
 E0M05640
 E0M05650

SUBROUTINE RESET (IERR)
 THE PURPOSE OF THIS ROUTINE IS: (1) TO RESET THE SEQUENCING OF
 DATA CARDS DUE TO AN ERROR IN A PREVIOUS MODULE OR (2) TO READ
 A SENTINEL CARD THAT DELINEATES THE END OF A DATA SET (SEE
 BELOW FOR DEFINITION OF A DATA SET) OR (3) TO STOP THE PROGRAM -
 THIS LAST MODE IS USUALLY FOR DEBUGGING OR TO ONLY CHANGE A CARD
 IN A COMPLETE RUN.
 TO DELINEATE THE END OF A DATA SET A CARD THAT HAS JUST END
 ON IT MUST BE INSERTED AS A SENTINEL CARD; A DATA SET IS
 DEFINED AS THAT COMPLETE SET OF CARDS NECESSARY TO RUN THE
 CALLED MODULE THE NUMBER OF TIMES AS SPECIFIED ON THE IDOPGM(I)
 CARD. STOP MAY ALSO BE INSERTED AS A SENTINEL CARD, IN WHICH
 CASE THE PROGRAM WILL BE TERMINATED AT THAT POINT - THIS IS
 NOT THE NORMAL TERMINATION.
 COMMON /IOUNIT/IOIN, IODOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOUTURES
 DATA ICHCK1, ICHCK2, ISTOP1, ISTOP2 /2HEN, 2HD, 2HST, 2HOP/
 IF(IERR.EQ.1) GO TO 1
 CONTINUE
 FOR UNIVAC AND IBM
 READ (IOIN, 100, END=2) ISNTL1, ISNTL2
 READ (IOIN, 100) ISNTL1, ISNTL2
 C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE
 C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.
 C3 READ (IOIN, 100) ISNTL1, ISNTL2
 C4 IF(EOF(IOIN)) 2,10
 10 IF((ISNTL1.NE.ICHCK1.AND.ISNTL2.NE.ICHCK2).AND.
 1 (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 5
 IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP
 RETURN
 1 WRITE (IODOUT, 102)
 C FOR UNIVAC AND IBM
 C6 READ (IOIN, 100, END=2) ISNTL1, ISNTL2
 READ (IOIN, 100) ISNTL1, ISNTL2
 C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE
 C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.
 C3 READ (IOIN, 100) ISNTL1, ISNTL2
 C4 IF(EOF(IOIN)) 2,20
 20 CONTINUE
 IF ((ISNTL1.NE.ICHCK1.AND.ISNTL.NE.ICHCK2).AND.
 1 (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 6
 IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP
 IERR=0
 RETURN
 2 WRITE (IODOUT, 101) IOIN
 100 FORMAT (2A2)
 101 FORMAT (1X, 120(1H*), /, 1X, 29H ERROR IN INPUT CONTROL FILE ,I4,
 + 21H - PROGRAM TERMINATED, /, 1X, 120(1H*))
 102 FORMAT (1H0, 50H**** CARD SEQUENCE RESET DUE TO ERROR IN PREVIOUS ,
 1 15HMODULE (IERR=1)//)

RES00010
 RES00020
 RES00030
 RES00040
 RES00050
 RES00060
 RES00070
 RES00080
 RES00090
 RES00100
 RES00110
 RES00120
 RES00130
 RES00140
 RES00150
 RES00160
 RES00170
 RES00180
 RES00190
 RES00200
 RES00210
 RES00220
 RES00230
 RES00240
 RES00250
 RES00260
 RES00270
 RES00280
 RES00290
 RES00300
 RES00310
 RES00320
 RES00330
 RES00340
 RES00350
 RES00360
 RES00370
 RES00380
 RES00390
 RES00400
 RES00410
 RES00420
 RES00430
 RES00440
 RES00450
 RES00460
 RES00470
 RES00480
 RES00490
 RES00500
 RES00510
 RES00520
 RES00530

```

SUBROUTINE COMPLT(LOTRNS,XSTRN,  

+                 SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN)      COM00010  

REAL LOTRNS,LZTRN,MMWTRN                               COM00020  

COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU COM00030  

LOWTRAN - LOTRNS                                     COM00050  

XSCALE - XSTRN                                      COM00060  

SMOKE - SMKTRN                                      COM00070  

DRTRAN - DRTRN                                      COM00080  

LZTRAN - LZTRN                                      COM00090  

NMMW - MMWTRN                                       COM00100  

CLTRAN - CLTRN                                      COM00110  

GRNADE - GRNTRN                                     COM00120  

TRAN=LOTRNS*XSTRN*SMKTRN*DRTRN*CLTRN*GRNTRN*MMWTRN*LZTRN   COM00130  

100  WRITE (I0OUT,100) TRAN                           COM00140  

      FORMAT (///1X,20X,24HCOMBINED TRANSMISSION F0,15HR THE SELECTED , COM00150  

      1 10HMODULES = ,E10.4)                          COM00160  

      RETURN                                           COM00170  

      END                                              COM00180

```

```

SUBROUTINE ILLUM(LAMBDA,LD,E0) ILL00010
REAL LAMBDA,LUNPHIA ILL00020
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU ILL00030
C***** ILL00040
C SUBROUTINE ILLUM RETURNS THE EXTRATERRESTRIAL IRRADIANCE E0 ILL00050
C AT WAVELENGTH LAMBDA. IF LD = 0, THE VALUE GIVEN IS SOLAR ILL00060
C IRRADIANCE. IF 1 < LD < 28 THE VALUE GIVEN IS LUNAR IRRAD- ILL00080
C IANCE ON LUNAR DAY LD, WITH DAY 28 CORRESPONDING TO FULL ILL00090
C MOON AND DAY 14 BEING NEW MOON. ILL00100
C SUBROUTINE COMPUTES VALUE OF LUNAR PHASE ANGLE, IF REQUIRED, ILL00110
C AND CALLS ONE OF THE EOSAEL ROUTINES SOLARS OR SMOON. ILL00120
C***** ILL00130
C IF(ILD.GT.0) GO TO 10 ILL00140
E0=SOLARS(LAMBDA) ILL00150
GO TO 100 ILL00160
C LD GT 0 => E0 = LUNAR VALUE ILL00170
C 10 ILD=LD ILL00180
IF(ILD.GT.14) ILD=28-ILD ILL00210
LUNPHIA=180.0*FLOAT(ILD)/14.0 ILL00220
E0=SMOON(LAMBDA,LUNPHIA) ILL00230
100 WRITE(1000) E0 ILL00240
1000 FORMAT(32H0 EXTRATERRESTRIAL IRRADIANCE= ,1PE10.4,11H W/M2-SR-MU)ILL00250
END ILL00260
ILL00270
ILL00280
ILL00290

```

```

FUNCTION SOLARS(WAVL)                               SOL00010
CALCULATE THE INTENSITY OF THE SOLAR SPECTRUM FOR WAVELENGTH (WAVL) SOL00020
SOL00030
UNITS:                                              SOL00040
SOL00050
      SOLARS ... WATTS M-2 MICRON-1               SOL00060
      WAVL    MICRONS                            SOL00070
IF (WAVL.LT.0.15.OR.WAVL.GE.100.) GO TO 100       SOL00080
IF (WAVL.GE.0.15.AND.WAVL.LE.0.43) GO TO 200       SOL00090
IF (WAVL.GT.0.43.AND.WAVL.LT.0.58) GO TO 300       SOL00100
GO TO 400                                         SOL00110
100 SOLARS=0.0                                     SOL00120
      RETURN                                         SOL00130
200 Z=((WAVL-.415)/0.68)**2                      SOL00140
      SOLARS=1775.*EXP(Z/2)
      RETURN                                         SOL00150
300 CONTINUE                                         SOL00160
      SOLARS=(-61142.*WAVL**4)+(1344477.*WAVL**3)-(110296.* SOL00170
      WAVL**2)
      SOLARS=(SOLARS+39952.7*WAVL-5371.)/100.
      RETURN                                         SOL00180
      SOL00190
400 IF (WAVL.GE.2.5) GO TO 500                     SOL00200
      SOLARS=5331.9*EXP(-1.959*WAVL)
      RETURN                                         SOL00210
500 SOLARS=2288.38*(WAVL**(-3.9765))             SOL00220
      RETURN                                         SOL00230
      END                                           SOL00240
                                                SOL00250
                                                SOL00260
                                                SOL00270

```

```

FUNCTION SMOON(WLAM,ANGLE)                               SMN00010
CALCULATE THE INTENSITY OF MOONLIGHT FOR WAVELENGTH (WLAM)   SMN00020
AND PHASE ANGLE (ANGLE)                                     SMN00030
UNITS:                                                 SMN00040
    SMOON ... WATTS M-2 MICRON-1                         SMN00050
    ANGLE ... DEGREES                                      SMN00060
    WLAM ... MICRONS                                     SMN00070
SMOON=0.0                                              SMN00080
IF (ANGLE.GT.160.) RETURN                                SMN00090
SMOON=(3.426E-9*ANGLE**4-1.63E-6*ANGLE**3+3.01E-4*      SMN00100
1     ANGLE**2-.0266*ANGLE+.9882)*100.                  SMN00110
ALBED=0.4                                              SMN00120
IF (WLAM.GE.5.) GO TO 200                                SMN00130
IF (WLAM.GT.2.8) GO TO 100                                SMN00140
IF (WLAM.LE.1.) ALBED=3.9633*WLAM**4-10.7306*WLAM**3+    SMN00150
1     10.2188*WLAM**2-3.8208*WLAM+.5512                 SMN00160
IF (WLAM.GT.1.) ALBED=.0482*WLAM**4-.3283*WLAM**3+      SMN00170
1     .7584*WLAM**2-.5745*WLAM+.2808                   SMN00180
GO TO 200                                              SMN00190
100 ALBED=.350+(-.500-.350)*(WLAM-2.8)/2.2            SMN00200
200 SMOON=2.04472E-07*SOLARS(WLAM)*ALBED*SMOON          SMN00210
RETURN                                                 SMN00220
END                                                   SMN00230
                                         SMN00240
                                         SMN00250

```

```
C      FUNCTION JPASCT(ICAT)
C      THIS FUNCTION CONVERTS THE INTEGER CODE FOR PASQUILL CATEGORY
C      TO THE ALPHA CHARACTER
C      DIMENSION NPASCT(6)
C      DATA NPASCT/1HA,1HB,1HC,1HD,1HE,1HF/
C      JPASCT=NPASCT(ICAT)
C      RETURN
C      END
```

JPA00010
JPA00020
JPA00030
JPA00040
JPA00050
JPA00060
JPA00070
JPA00080

```

SUBROUTINE PFUNC(IDN) PFU00010
LOGICAL ISPOT, LREAD, N16 PFU00020
DIMENSION PFSPOT(16), PFH(65) PFU00030
COMMON /SPOTLO/ISPOT, LREAD, N16 PFU00040
COMMON /IOUNIT/IOUT, IOIN, IPHFUN, LOUNIT, HDIRTU, NCLIMT, KSTOR, NPLOTU PFU00050
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDECK PFU00060
COMMON /CGEOM/COSGM, COSBT, COSIN PFU00070
COMMON /BASBOT/ANG(65), SUM(65), WVL(16), NWVL, ALB(16), BSC(16), PFU00080
+BE(16), SINGWV, PF(65), LMAX PFU00090
PFU00100
C THIS SUBROUTINE PERFORMS PHASE FUNCTION READING AND INTERPOL- PFU00110
ATION OPERATIONS FOR THE SPOT AND BASCAT MODULES. THE FILE IN- PFU00120
WHICH THE PHASE FUNCTION DATA RESIDES IS PFNDAT. USERS MAY IN- PFU00130
SERT PHASE FUNCTIONS OF THEIR OWN SPECIFICATION INTO PFNDAT PFU00140
UNDER AN ID NUMBER 0. THIS PHASE FUNCTION MAY HAVE ANY AR- PFU00150
BITRARY NORMALIZATION, SINCE PFUNC WILL RENORMALIZE IT TO PFU00160
CONFORM TO THE NORMALIZATIONS USED IN SPOT AND BASCAT. PFU00170
FOR FURTHER DETAILS ON THIS PROCEDURE PFU00180
THE USER IS REFERRED TO CHAPTER 16 OF THE EOSAEL 80 TECHNICAL PFU00190
DOCUMENTATION MANUAL, WHERE THE STRUCTURE, USE, AND MODIFICATION PFU00200
OF THE PFNDAT FILE IS DISCUSSED. PFU00210
PFU00220
C*** NOTE *** FOR USER-DEFINED PHASE FUNCTIONS (IDN=0), THIS ROUTINE PFU00230
WILL INTERPOLATE OVER WAVELENGTH AND ANGLE FOR THE SPOT PFU00240
MODULE. FOR THE BASCAT MODULE, HOWEVER, NO WAVELENGTH PFU00250
INTERPOLATION IS PERFORMED FOR THE USER-DEFINED PHASE PFU00260
FUNCTION. ONLY RENORMALIZATION IS PERFORMED IN THIS PFU00270
LATTER CASE. PFU00280
PFU00290
C*** MAXID IS THE NUMBER OF DIFFERENT DISTRIBUTIONS - (PHASE FUNCTIONS) PFU00300
C MAXID=12 PFU00310
PFU00320
PFU00330
C*** CHECK THE ALLOWABLE RANGE OF DISTRIBUTIONS PFU00340
C IF((IDN.GT.MAXID).OR.(IDN.LT.0))GO TO 491 PFU00350
PFU00360
PFU00370
C*** DATA INITIALIZATIONS PFU00380
PFU00390
C DO 40 I=1,65 PFU00400
40 PF(I)=0, PFU00410
ALBE=0, PFU00420
BEX=0, PFU00430
BSC=0, PFU00440
PFU00450
C*** ANGULAR READ BLOCK COMMON TO SPOT AND BASCAT PFU00460
C I=0 PFU00470
PFU00480
C**** READ IN ANGLES AT WHICH PHASE FUNCTION IS DEFINED PFU00490
PFU00500
C 50 I=I+1 PFU00510
L1=(I-1)*11+1 PFU00520
L11=L1+10 PFU00530
IF(L11.EQ.66)L11=65 PFU00540
IF(L11.GT.66)GO TO 492 PFU00550
READ(IPHFUN,60)(ANG(L),L=L1,L11) PFU00560
60 FORMAT(11(F6.2,1X)) PFU00570
PFU00580
C**** CHECK THIS ROW OF DATA FILE FOR TERMINATION SENTINEL PFU00590
PFU00600
C DO 100 K=L1,L11 PFU00610
IF((ANG(K).GE.999.99).AND.(L11.LT.65))GO TO 200 PFU00620
100 IF(K.EQ.65)GO TO 200 PFU00630
CONTINUE PFU00640
GO TO 50 PFU00650
PFU00660
C*** LMAX IS THE NBR OF ANGLE AND NBR OF PHASE FUNCTION VALUES+1 HERE. PFU00670
200 LMAX=K PFU00680
PFU00690
PFU00700

```

```

      DO 250 L=1,LMAX          PFU00710
250  ANG(L)=COS(ANG(L)*PIRAD) PFU00720
C***  REDUCE DUE TO SENTINEL OF 999.99 PFU00730
C
      IF(L11.LT.65)LMAX=K-1 PFU00740
      IDNM=IDN-1 PFU00750
      KMAX=IFIX ALOG(FLOAT(LMAX-1))/ALOG(2.0)+0.1) PFU00760
PFU00770
PFU00780
PFU00790
PFU00800
PFU00810
PFU00820
PFU00830
PFU00840
PFU00850
PFU00860
PFU00870
PFU00880
PFU00890
PFU00900
PFU00910
PFU00920
PFU00930
PFU00940
PFU00950
PFU00960
PFU00970
PFU00980
PFU00990
PFU01000
PFU01010
PFU01020
PFU01030
PFU01040
PFU01050
PFU01060
PFU01070
PFU01080
PFU01090
PFU01100
PFU01110
PFU01120
PFU01130
PFU01140
PFU01150
PFU01160
PFU01170
PFU01180
PFU01190
PFU01200
PFU01210
PFU01220
PFU01230
PFU01240
PFU01250
PFU01260
PFU01270
PFU01280
PFU01290
PFU01300
PFU01310
PFU01320
PFU01330
PFU01340
PFU01350
PFU01360
PFU01370
PFU01380
PFU01390
PFU01400
C***  RESET PARAMETERS FOR BASCAT PROCESSING IF APPROPRIATE.
C
      IF(ISPOT) GO TO 260
      NWVL=1
      WVL(1)=SINGWV
260  CONTINUE
      IF((IDN.EQ.1).OR.(IDN.EQ.0.)) GO TO 1050
C***  READ PAST AEROSOL DATA NOT OF CURRENT INTEREST
C
      DO 1000 I=1, IDNM
      DO 1000 II=1,16
      READ(IPHFUN,300) IANG, ID, WAVE, ALBE, BEX, BSC
300  FORMAT(2(E12.6),F5.2,1X,F8.6,1X,2(E12.6,1X))
      IF(IANG.NE.LMAX) GO TO 493
      READ(IPHFUN,400) (PF(L),L=1,LMAX)
400  FORMAT(6(E12.6,1X))
1000 CONTINUE
1050 CONTINUE
C***  OMIT WAVELENGTH CHECKS FOR USER-DEFINED PHASE FUNCTION.
C
      IF(IDN.EQ.0) GO TO 1070
C***  THE NEXT LOOP PERFORMS THE FOLLOWING OPERATIONS :
      (A) IT VERIFIES WHETHER OR NOT ALL INPUT WAVELENGTHS LIE
          WITHIN THE 0.2-12.0 MICROMETER RANGE (WITH LIMITS
          EXTENDED TO PLUS OR MINUS 5%).
      (B) IF THE PHASE FUNCTION IS NOT USER-SPECIFIED AND THE INPUT
          WAVELENGTH BAND LIES WITHIN THE 0.2-2.0 BAND, THEN
          INTERPOLATION IS NOT POSSIBLE DUE TO THE PRESENCE OF
          ONLY TWO DATA POINTS (AT 0.55 AND 1.06) IN THIS REGION.
          AS A RESULT, THE 0.55 AND 1.06 VALUES ARE ASSIGNED TO
          INDIVIDUAL POINTS IN THE INPUT WAVELENGTH BAND. THOSE
          POINTS WITH WAVELENGTH VALUES LESS THAN OR EQUAL TO 0.8
          MICROMETERS ARE ASSIGNED THE 0.55 DATA. ALL OTHER WAVE-
          LENGTHS ARE ASSIGNED THE 1.06 DATA.
      (C) IF AN INPUT WAVELENGTH LIES OUTSIDE OF THE 3-5 OR 8-12
          MICROMETER BANDS, BUT IS WITHIN 5% OF AN EXTREMUM FOR
          THESE BANDS, IT IS RESET TO THE EXTREMUM WAVELENGTH
          VALUE.
      (D) IF AN INPUT WAVELENGTH LIES BETWEEN BANDS, EXECUTION IS
          TERMINATED AND AN ERROR MESSAGE IS PRINTED.
C
      DO 1060 I=1,NWVL
      IF((WVL(I).LT.0.19).OR.(WVL(I).GT.12.049)) GO TO 504
      IF(WVL(I).LE.0.8) WVL(I)=0.55
      IF((WVL(I).GT.0.8).AND.(WVL(I).LE.2.1)) WVL(I)=1.06
      IF(WVL(I).GT.12.0) WVL(I)=12.0
      IF((WVL(I).GT.5.0).AND.(WVL(I).LE.5.25)) WVL(I)=5.0
      IF((WVL(I).GE.7.6).AND.(WVL(I).LT.8.0)) WVL(I)=8.0
      IF((WVL(I).GE.2.85).AND.(WVL(I).LT.3.0)) WVL(I)=3.0
      IF((WVL(I).GT.2.1).AND.(WVL(I).LT.2.85)) GO TO 498
      IF((WVL(I).GT.5.25).AND.(WVL(I).LT.7.6)) GO TO 498
1060 CONTINUE
1070 CONTINUE
C***  MAIN INTERPOLATION LOOP
C
      DO 2000 I=1,NWVL
      IF(I.GT.1) GO TO 1260

```

```

1100 CONTINUE PFU01410
    READ(IPHFUN,300) IANG, ID, WAVEH, ALBH, BEXH, BSCH PFU01420
    READ(IPHFUN,400) (PFH(L), L=1,LMAX) PFU01430
    IF((IDN,EQ,0),AND,(.,NOT.ISPOT)) GO TO 1280 PFU01440
    IF(WVL(I).LT.WAVEH) GO TO 1100 PFU01450
1150 CONTINUE PFU01460
    READ(IPHFUN,300) IANG, ID, WAVE, ALBE, BEX, BSC PFU01470
    READ(IPHFUN,400) (PF(L), L=1,LMAX) PFU01480
    IF(WVL(I).LE.WAVE) GO TO 1260 PFU01490
1160 CONTINUE PFU01500
    WAVEH=WAVE PFU01510
    ALBH=ALBE PFU01520
    BEXH=BEX PFU01530
    DO 1240 L=1,LMAX PFU01540
1240 PFH(L)=PF(L) PFU01550
    GO TO 1150 PFU01560
1260 CONTINUE PFU01570
C PFU01580
C*** GO TO NEXT WAVELENGTH INTERPOLATION INTERVAL IF INPUT WAVE- PFU01590
C*** LENGTH IS GREATER THAN THE MAXIMUM OF THE CURRENT ONE. PFU01600
C PFU01610
C IF(WVL(I).GT.WAVE) GO TO 1160 PFU01620
C PFU01630
C*** RENORMALIZE LOWER END OF INTERPOLATION INTERVAL PFU01640
C PFU01650
1280 CONTINUE PFU01660
    SUM(1)=0.0 PFU01670
    DO 1200 L=2,LMAX PFU01680
1200 SUM(L)=((ANG(L-1)-ANG(L))*(PFH(L-1)+PFH(L))/4.0)+SUM(L-1) PFU01690
    SUMT=SUM(LMAX) PFU01700
    DO 1250 L=1,LMAX PFU01710
1250 PFH(L)=PFH(L)/SUMT PFU01720
C PFU01730
C*** BRANCH TO FINAL PROCEDURE FOR BASCAT USER-DEFINED PHASE PFU01740
C*** FUNCTION IF APPROPRIATE. PFU01750
C PFU01760
C IF((IDN,EQ,0),AND,(.,NOT.ISPOT)) GO TO 2500 PFU01770
C PFU01780
C*** RENORMALIZE UPPER END OF INTERPOLATION INTERVAL. PFU01790
C PFU01800
    SUM(1)=0.0 PFU01810
    DO 1400 L=2,LMAX PFU01820
1400 SUM(L)=((ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0)+SUM(L-1) PFU01830
    SUMT=SUM(LMAX) PFU01840
    DO 1450 L=1,LMAX PFU01850
1450 PF(L)=PF(L)/SUMT PFU01860
C PFU01870
C*** BRANCH TO BASCAT WAVELENGTH INTERPOLATION PROCEDURE IF PFU01880
C*** APPROPRIATE. PFU01890
C PFU01900
C IF(.NOT.ISPOT) GO TO 2500 PFU01910
C PFU01920
C*** PERFORM HALVING SEARCH FOR COSINES IN PHASE FUNCTION DATA FILE PFU01930
C*** WHICH BRACKET COSINE INPUT FROM SPOT. PFU01940
C PFU01950
    L=1 PFU01960
    LL=LMAX-1 PFU01970
    DO 1300 K=1,KMAX PFU01980
    LL=LL/2 PFU01990
    L=L+LL PFU02000
    AT=COSIN-ANG(L) PFU02010
    IF(AT.GT.0.) L=L-LL PFU02020
1300 CONTINUE PFU02030
C PFU02040
C*** PERFORM SPOT EXTINCTION COEFFICIENT AND PHASE FUNCTION PFU02050
C*** INTERPOLATIONS OVER WAVELENGTH AND ANGLE. PFU02060
C PFU02070
    FACANG=(COSIN-ANG(L))/(ANG(L+1)-ANG(L)) PFU02080
    FACWVL=(WVL(I)-WAVEH)/(WAVE-WAVEH) PFU02090
    BE(I)=BEXH+(BEX-BEXH)*FACWVL PFU02110

```

```

PFSPOT(I)=PFH(L)*(1.-FACANG-FACWVL+FACANG*FACWVL)+  

+PFH(L+1)*(FACANG-FACANG*FACWVL)+PF(L)*(FACWVL-FACANG*FACWVL)+  

+PF(L+1)*(FACANG*FACWVL)
2000 CONTINUE
C*** LOAD FIRST NWVL ANGLES OF OUTPUT ARRAY PF() WITH INTERPOLATED
C*** RESULTS FOR SPOT.
C
DO 2200 N=1,NWVL
2200 PF(N)=PFSPOT(N)/(4.*PI)
C*** FINAL EXIT FOR SPOT PROCESSING.
C
GO TO 500
2500 CONTINUE
FACWVL=0.
IF(IDN.EQ.0) GO TO 2700
C*** BASCAT ALBEDO, EXTINCTION COEFFICIENT, AND PHASE FUNCTION
C*** INTERPOLATION OVER WAVELENGTH.
C
FACWVL=(WVL(1)-WAVEH)/(WAVE-WAVEH)
2700 CONTINUE
DO 2800 L=1,LMAX
2800 PF(L)=PFH(L)+(PF(L)-PFH(L))*FACWVL
ALB(1)=ALBH+(ALBE-ALBH)*FACWVL
BE(1)=BEXH+(BEX-BEXH)*FACWVL
C*** FINAL EXIT FOR BASCAT USER-DEFINED PHASE FUNCTION PROCEDURE.
C
IF(IDN.EQ.0) GO TO 500
C*** FINAL BASCAT PHASE FUNCTION RENORMALIZATION.
C
SUM(1)=0.0
DO 2900 L=2,LMAX
2900 SUM(L)=(ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0+SUM(L-1)
SUMT=SUM(LMAX)
DO 2950 L=1,LMAX
2950 PF(L)=PF(L)/SUMT
C*** FINAL EXIT FOR BASCAT PROCESSING.
C
GO TO 500
C*** ERROR EXIT BLOCK COMMON TO SPOT AND BASCAT
C
491 CONTINUE
WRITE(IOUT,495)
495 FORMAT(1H0,20X,58H***PFUNC ERROR*** AEROSOL ID NUMBER OUT OF ALLOWPFU02610
+ABLE RANGE //)
STOP
492 CONTINUE
WRITE(IOUT,496)
496 FORMAT(1H0,20X,83H***PFUNC ERROR*** READ TERMINATION SENTINEL NOT PFU02660
+FOUND OR NUMBER OF ANGLES EXCEED 65 //)
STOP
493 CONTINUE
WRITE(IOUT,497)
497 FORMAT(1H0,20X,93H***PFUNC ERROR*** NUMBER OF SPECIFIED ANGLES ANDPFU02710
+ NUMBER OF PHASE FUNCTION VALUES DO NOT MATCH //)
STOP
498 CONTINUE
WRITE(IOUT,499)
499 FORMAT(1H0,10X,79H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WYPFU02760
+L INPUT ARRAY DO NOT LIE WITHIN /1H ,45HWAVELENGTH BANDS COVERED BPFU02770
+Y PFNDAT DATA BASE /)
STOP
504 CONTINUE
WRITE(IOUT,505)

```

505 FORMAT(1H0,10X,109H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WVPFU02820
+L ARRAY DO NOT LIE WITHIN OVERALL ACCEPTABLE RANGE OF 0.2-12.0 /> PFU02830
STOP PFU02840
500 RETURN PFU02850
END PFU02860

SUBROUTINE XSCALE(WAVE,VIS,EXT55,XSTRN,IERR,ISLT,IFOG,RANGE,ANGLE)XSC00010
 THE PURPOSE OF THIS ROUTINE IS TO A) FIND THE HORIZONTAL EXTINCTIONXSC00020
 IN FOG AT THE WAVELENGTHS SPECIFIED BELOW FROM THE EXTINCTIONXSC00030
 AT .55 UM OR B) THE EXTINCTION ALONG A SLANT PATH AT ALLOWEDXSC00040
 WAVELENGTHS FROM THE EXTINCTION AT .55 UM: FOG TYPE 1, 2, OR 3, XSC00050
 MUST BE SPECIFIED FOR SLANT PATHS. XSC00060
 *** VISIBILITY = 88. OR 89. IS NOT ALLOWED AS THIS IS USED AS ANXSC00070
 INDICATOR THAT XSCALE IS BEING CALLED AS A SUBROUTINE FROM EITHERXSC00080
 SPOT(88.), LOWTRAN(88.), OR CWIC(89.), NOT EOMAIN!XSC00090
 WAVE=LAMDA IN UM - MUST BE .55, 1.06, 3.0-5.0, 8.0-12.05. XSC00100
 *** ALL EXTN'S ARE IN KM***-1 XSC00110
 EXT55 = EXTINCTION AT .55 UM XSC00120
 EXT106 = EXTINCTION AT 1.06 UM XSC00130
 EXT35 = EXTINCTION FROM 3.0 TO 5.0 UM XSC00140
 EXT812 = EXTINCTION FROM 8.0 TO 12.0 UM XSC00150
 VIS= VISIBILITY IN KM -OR- EXT55 IN KM***-1 XSC00160
 EXT55 IS ** NOT ** CHANGED BY THIS ROUTINE. XSC00170
***** INPUT: THERE IS A MAXIMUM OF 3 CARDS TO EXECUTE THIS MODULE XSC00180
 THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OF XSC00190
 THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN. XSC00200
 THE CARDS ARE INPUT WITH FORMAT (A4,6X,5(F6.2,1X)). XSC00210
 EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4 XSC00220
 FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 6 COL PER XSC00230
 FIELD BEGINNING IN COL 11, WITH A BLANK BETWEEN EACH SUBSEQUENT XSC00240
 FIELD. THE CARDS ARE NOT ORDER DEPENDENT. XSC00250
 IF GEOMET OPTION IS BEING USED, THEN ONLY THE IDENTIFIER HORZ, XSC00260
 SLNH, OR SLNS IS TO BE READ IN <NO ADDITIONAL PARAMETERS NEEDED>,
 FOG FOG TYPE, RAIN RATE <MM/HR>; RAIN RATE ONLY NEEDED XSC00270
 WHEN FOG TYPE=4. XSC00280
 HORZ HORDIS <KM> :HORIZONTAL PATH CALCULATION XSC00300
 SLNH HORDIS <KM>; ANGLE <DEGREES>; SLANT PATH CALCULATION XSC00310
 SLNS SLTDIS <KM>; ANGLE <DEGREES>; SLANT PATH CALCULATION XSC00320
 PLOT WRITE SLANT PATH EXTINCTION, AT INPUT WAVELENGTH, AND
 ALTITUDE TO NPLTU (SEE COMMON BLOCK IOUNIT); THE
 FIRST RECORD WILL BE THE NUMBER OF POINTS TO BE WRITTEN,
 FORMATS: RECORD 1 (I5), SUBSEQUENT RECORDS (2(E10.4,1X)) XSC00330
 GO SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FOR XSC00340
 THIS CALL. NOTE THAT IF A DATA CARD IS NOT READ XSC00350
 THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLS XSC00360
 TO THE MODULE ARE STILL IN EFFECT.
 ALL THE FOLLOWING FOG TYPES ARE RELEVANT TO HORIZONTAL PATHS,
 BUT ONLY FOG TYPES 1, 2, OR 3 ARE ALLOWED FOR SLANT PATH CALCULATIONS XSC00380
 FOG TYPE=1. FOR MARITIME ARTIC
 =2. FOR MARITIME POLAR XSC00390
 =3. FOR CONTINENTAL POLAR XSC00400
 =4. FOR RAIN XSC00410
 =5. FOR SNOW XSC00420
 HORDIS - HORIZONTAL DISTANCE IN KM. XSC00430
 SLTDIS - SLANT PATH DISTANCE IN KM. XSC00440
 ANG= LOOK ANGLE FROM HORIZONTAL IN DEGREES XSC00450
 N.B. ONE OF THE FOLLOWING COMBINATIONS MUST BE SUPPLIED XSC00460
 FOR SLANT PATH CALCULATIONS. XSC00470
 HORDIS AND ANG ** OR ** SLTDIS AND ANG XSC00480
***** OUTPUT XSC00490
 TRANSMISSION AT APPROPRIATE WAVELENGTH FOR SLANT OR HORIZONTAL PATHXSC00510
 COMMON /CONST/PI,PI2,PIRAD,TWOP,PIRMB,CDEGK XSC00530
 COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NPLTU XSC00540
 COMMON /GEOMET/PTS(15),IGEOSW XSC00550
 DIMENSION TYPE(6),DAT(6) XSC00560
 LOGICAL N0LD XSC00570
 DATA TYPE /4HFOG,4HHORZ,4HSLNH,4HSLNS,4HPLOT,4HGO / XSC00580
 DATA A0,A1,A2,NPLT/0.1425,0.1475,-0.0017,0/ XSC00590
 USE VIS=88. OR 89. AS AN INDICATOR THAT XSCALE HAS BEEN CALLED XSC00600
 AS A SUBROUTINE FROM OTHER PROGRAMS - NOT EOMAIN! XSC00610
 IF (VIS.LT.87.9.OR.VIS.GT.89.1) GO TO 8 XSC00620
 ANG=ANGLE XSC00630
 FIND ELEVATION ANGLE FROM ZENITH ANGLE IN SPOT AND LOWTRAN XSC00640

```

IF <VIS.GT.87.9.AND.VIS.LT.88.1> ANG=90.-ANGLE XSC00650
IF <VIS.GT.87.9.AND.VIS.LT.88.1.AND.ANGLE.GT.90.> ANG=ANGLE-90. XSC00660
HORDIS=0, XSC00670
SLTDIS=0, XSC00680
NULU=.FALSE. XSC00690
C ISLT=0=HORIZONTAL; ISLT GT 0 = SLANT. XSC00700
IF <ISLT.EQ.0> HORDIS=RANGE XSC00710
IF <ISLT.EQ.0> ISLANT=0 XSC00720
IF <ISLT.GT.0> SLTDIS=RANGE XSC00730
IF <ISLT.GT.0> ISLANT=1 XSC00740
GO TO 6 XSC00750
8 CONTINUE XSC00760
NULU=.TRUE. XSC00770
DO 9 I=1,3 XSC00780
READ <IOIN,500> <DAT(J),J=1,5> XSC00790
IF <DAT(1).EQ.TYPE(1)> GO TO 1 XSC00800
IF <DAT(1).EQ.TYPE(2)> GO TO 2 XSC00810
IF <DAT(1).EQ.TYPE(3)> GO TO 3 XSC00820
IF <DAT(1).EQ.TYPE(4)> GO TO 4 XSC00830
IF <DAT(1).EQ.TYPE(5)> GO TO 5 XSC00840
IF <DAT(1).EQ.TYPE(6)> GO TO 6 XSC00850
C ERROR CHECK XSC00860
GO TO ? XSC00870
C ADVERSE WEATHER INDICATOR AND OPTIONAL RAIN RATE: XSC00880
1 IFOG=IFIX(DAT(2)) XSC00890
RHRT=DAT(3) XSC00900
GO TO 9 XSC00910
C HORIZONTAL DISTANCE FOR HORIZONTAL PATH CALC. XSC00920
2 HORDIS=DAT(2) XSC00930
ISLANT=0 XSC00940
GO TO 9 XSC00950
C HORIZONTAL DISTANCE AND ANGLE FOR SLANT PATH CALC. XSC00960
3 HORDIS=DAT(2) XSC00970
ANG=DAT(3) XSC00980
ISLANT=1 XSC00990
GO TO 9 XSC01000
C SLANT DISTANCE AND ANGLE FOR SLANT PATH CALC. XSC01010
4 SLTDIS=DAT(2) XSC01020
ANG=DAT(3) XSC01030
ISLANT=1 XSC01040
GO TO 9 XSC01050
C SET PLOT FLAG XSC01060
5 NPLT=1 XSC01070
CONTINUE XSC01080
6 CONTINUE XSC01090
IF <NOLO> WRITE <IOOUT,600> XSC01100
IF <ICEOSW.NE.1> GO TO 88 XSC01110
HORDIS=SQRT(<PTS(1)-PTS(4)>**2+<PTS(2)-PTS(5)>**2) XSC01120
SLTDIS=SQRT(HORDIS**2+<PTS(3)-PTS(6)>**2) XSC01130
ANG=ACOS(HORDIS/SLTDIS)/PIRAD XSC01140
88 CONTINUE XSC01150
C WAVELENGTH ERROR CHECK XSC01160
IF <WAVE.GT..4.AND.WAVE.LE.2.,.OR.<WAVE.GE.3..AND.WAVE.LE.5.,> XSC01170
1,.OR.<WAVE.GE.8.AND.WAVE.LE.12.05> GO TO 10 XSC01180
WRITE <IOOUT,1600> WAVE XSC01190
IERR=1 XSC01200
XSTRN=1. XSC01210
RETURN XSC01220
10 CONTINUE XSC01230
IF <NOLO.AND.IFOG.EQ.1> WRITE <IOOUT,800> XSC01240
IF <NOLO.AND.IFOG.EQ.2> WRITE <IOOUT,900> XSC01250
IF <NOLO.AND.IFOG.EQ.3> WRITE <IOOUT,950> XSC01260
IF <NOLO.AND.IFOG.EQ.4> WRITE <IOOUT,1000> XSC01270
IF <NOLO.AND.IFOG.EQ.5> WRITE <IOOUT,1100> XSC01280
IF <ISLANT.GT.0> GO TO 11 XSC01290
IF <NOLO> WRITE <IOOUT,1200> XSC01300
11 IF <NOLO.AND.ISLANT.GE.1> WRITE <IOOUT,1400> WAVE XSC01310
IF <ISLANT.GE.1.AND.<IFOG.LE.0.OR.IFOG.GE.4>> WRITE <IOOUT,2100> XSC01320
IF <ISLANT.GE.1.AND.<IFOG.LE.0.OR.IFOG.GE.4>> IFOG=1 XSC01330
EXTN=EXT55

```

```

12   IF (ISLANT.GE.1.AND.WAVE.GE.,8) GO TO 101 XSC01280
    1 IF (ISLANT.GE.1) CALL SLANT(EXTN,HORDIS,SLTDIS,ANG,AVEX55,IERR, XSC01290
      WAVE,NPLT) XSC01300
    IF (IERR.EQ.1) XSTRN =1. XSC01310
    IF (IERR.EQ.1) RETURN XSC01320
    EXTN=EXT55 XSC01330
    IF (ISLANT.GE.1) EXTN=AVEX55 XSC01340
    XSPATH=HORDIS XSC01350
    IF (ISLANT.GE.1) XSPATH=SLTDIS XSC01360
    IF (ISLANT.EQ.0) GO TO 100 XSC01370
C     SLANT PATH EXTINCTION XSC01380
    XSTRN =EXP(-XSPATH*EXTN) XSC01390
    IF (NOLO) WRITE (IO00UT,1500) WAVE,EXTN,XSPATH,XSTRN,ANG XSC01400
    RETURN XSC01400
100  IF (IFOG.NE.4) GO TO 101 XSC01410
C     RAIN - ALL WAVELENGTHS XSC01420
    IF (NOLO.AND.RNRT.LE.0.0) WRITE (IO00UT,550) XSC01430
    IF (RNRT.LE.0.0) RNRT=1. XSC01440
    RNEXTN=A0+A1*RNRT+A2*RNRT**2 XSC01450
    XSTRN =EXP(-XSPATH*RNEXTN) XSC01460
    IF (NOLO) WRITE (IO00UT,1550) RNRT,RNEXTN,XSPATH,XSTRN XSC01470
    RETURN XSC01470
101  IF (ABS(WAVE-1.06).LT..01) GO TO 400 XSC01480
    IF (WAVE.GE.3.0.AND.WAVE.LE.5.0) GO TO 200 XSC01490
    IF (WAVE.GE.8.0.AND.WAVE.LE.12.0) GO TO 300 XSC01500
    XSTRN=EXP(-XSPATH*EXTN) XSC01510
    RETURN XSC01520
200  CONTINUE XSC01530
C     3.0 TO 5.0 RANGE XSC01540
C     MA XSC01550
C     MP XSC01560
C     IF (IFOG.EQ.2) EXT35=10.**(-0.38+1.32*ALOG10(EXTN)) XSC01570
C     CP XSC01580
C     IF (IFOG.EQ.3) EXT35=10.**(-0.82+1.58*ALOG10(EXTN)) XSC01590
C     IF (IFOG.NE.5) EXTN=EXT35 XSC01600
C     IF (ISLANT.GE.1) GO TO 12 XSC01610
C     SNOW XSC01620
C     IF (IFOG.EQ.5) EXT35=10.0**(.105*ALOG10(EXTN)+.021) XSC01630
    XSTRN =EXP(-XSPATH*EXT35) XSC01640
    IF (NOLO) WRITE (IO00UT,1700) EXT35,XSPATH,XSTRN XSC01650
    RETURN XSC01660
300  CONTINUE XSC01670
C     8.0 TO 12.0 RANGE XSC01680
C     MA XSC01690
C     MP XSC01700
C     IF (IFOG.EQ.2) EXT812=10.**(-1.01+1.51*ALOG10(EXTN)) XSC01710
C     CP XSC01720
C     IF (IFOG.EQ.3) EXT812=10.**(-1.65+1.82*ALOG10(EXTN)) XSC01730
C     IF (IFOG.NE.5) EXTN=EXT812 XSC01740
C     IF (ISLANT.GE.1) GO TO 12 XSC01750
C     SNOW XSC01760
C     IF (IFOG.EQ.5) EXT812=10.0**(.993*ALOG10(EXTN)+.114) XSC01770
    XSTRN =EXP(-XSPATH*EXT812) XSC01780
    IF (NOLO) WRITE (IO00UT,1800) EXT812,XSPATH,XSTRN XSC01790
    RETURN XSC01800
400  CONTINUE XSC01810
C     1.06 RANGE XSC01820
C     MA, MP, AND CP XSC01830
    EXT106=AMIN1(10.**(-0.14+1.16*ALOG10(EXTN)),EXTN) XSC01840
    EXTN=EXT106 XSC01850
    IF (ISLANT.GE.1) GO TO 12 XSC01860
    SNOW - ASSUME THAT THE EXTINCTION AT 1.06 IS THE SAME AS AT .55 UMXSC01870
    IF (IFOG.EQ.5) EXT106=EXT55 XSC01880
    XSTRN =EXP(-XSPATH*EXT106) XSC01890
    IF (NOLO) WRITE (IO00UT,1900) EXT106,XSPATH,XSTRN XSC01900
    RETURN XSC01910
7     WRITE (IO00UT,2000) (DAT(J),J=1,4) XSC01920
    XSTRN=1. XSC01930

```

```

IERR=1          XSC01940
RETURN         XSC01950
C
500  FORMAT (A4,6X,5(F6.2,1X))          XSC01960
550  FORMAT(1H0,50HXSCALE WARNING - RAIN RATE IS .LE. 0.0, RAIN RATE , XSC01970
      +20HSET EQUAL TO 1 MM/HR,)          XSC01980
600  FORMAT (1H ,//,,1X,50X,14HOPTIONS CHOSEN) XSC01990
800  FORMAT (1H ,50X,14HMARITIME ARTIC)      XSC02000
900  FORMAT (1H ,50X,14HMARITIME POLAR)      XSC02010
950  FORMAT (1H ,50X,17HCONTINENTAL POLAR)   XSC02020
1000 FORMAT (1H ,50X,4HRAIN)                XSC02030
1100 FORMAT (1H ,50X,4HSNOW)               XSC02040
1200 FORMAT (1H ,50X,15HHORIZONTAL PATH/)   XSC02050
1400 FORMAT (1H ,50X,15HSLANT PATH FOR ,F8.3,8H MICRONS/) XSC02060
1500 FORMAT (1X,23X,20HSLANT EXTINCTION AT ,F5.2,8H MICRONS, XSC02070
      1   4X,11HDISTANCE ,12HTRANSMISSION,6X,5HANGLE/,1X,40X, XSC02080
      2   26HKM**-1   KM,/                   XSC02090
      3   ,1X,38X,F8.3,12X,F8.3,5X,E9.3,7X,F7.2) XSC02100
1550 FORMAT (1H ,20X,27HEXTINCTION FOR RAIN RATE OF ,F5.2,6H MM/HR, XSC02110
      1   5X,11HDISTANCE ,12HTRANSMISSION/,1X,40X, XSC02120
      2   26HKM**-1   KM,/                   XSC02130
      3   ,1X,38X,F8.3,12X,F8.3,5X,E9.3) XSC02140
1600 FORMAT (1H ,18H***** WAVELENGTH ( ,F9.3,10H) OUTSIDE XSC02150
      1   10HALLOWABLE ,29HRANGE (1.06,3.0-5.0,8.0-12.0 XSC02160
      2   31HMICRONS) - CONTROL RETURNED TO , XSC02170
      3   17HMAIN FROM XSCALE,) XSC02180
1700 FORMAT (1H ,25X,37HEXTINCTION FROM 3.0 TO 5.0 MICRONS XSC02190
      1   11HDISTANCE ,12HTRANSMISSION/,1X,40X, XSC02200
      2   26HKM**-1   KM,/                   XSC02210
      3   ,1X,38X,F8.3,13X,F8.3,5X,E9.3) XSC02220
1800 FORMAT (1H ,25X,37HEXTINCTION FROM 8.0 TO 12.0 MICRONS XSC02230
      1   12H DISTANCE ,12HTRANSMISSION/,1X,40X, XSC02240
      2   26HKM**-1   KM,/                   XSC02250
      3   ,1X,38X,F8.3,12X,F8.3,5X,E9.3) XSC02260
1900 FORMAT (1H ,25X,37HEXTINCTION AT 1.06 MICRONS XSC02270
      1   11HDISTANCE ,12HTRANSMISSION/,1X,40X, XSC02280
      2   26HKM**-1   KM,/                   XSC02290
      3   ,1X,38X,F8.3,12X,F8.3,5X,E9.3) XSC02300
2000 FORMAT (1H ,44HUNKNOWN CARD TYPE, CONTROL RETURNED TO MAIN, XSC02310
      +13H FROM XSCALE,/,1X,A4,6X,5(F6.2,1X)) XSC02320
2100 FORMAT(1H ,40HINCORRECT FOG TYPE FOR SUBROUTINE SLANT,,/,1X, XSC02330
      1   21HFog TYPE CHANGED TO 1/) XSC02340
END

```

```

SUBROUTINE SLANT(EXT55,HORDIS,SLTDIS,ANG,AVEX55,IERR,WAVE,NPLT)      SLNT0010
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSLNT0020
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                          SLNT0030
C   ALL QUANTITIES IN THIS ROUTINE ARE FOR .55 UM                         SLNT0040
C   REAL KMTOM                SLNT0050
C   KILOMETERS TO METERS        SLNT0060
C   KMTOM=1000.                  SLNT0070
C   IERR=0                      SLNT0080
C   TOL=.0001                   SLNT0090
C 100 IF (HORDIS.LT.TOL.OR.ANG.LT.TOL) GO TO 200                      SLNT0100
C   HORIZONTAL DISTANCE AND ANGLE INPUT                                SLNT0110
C   VERDIS=HORDIS*TAN(ANG*PIRAD)                                         SLNT0120
C   SLTDIS=SQRT(HORDIS**2+VERDIS**2)                                       SLNT0130
C   GO TO 300                SLNT0140
C 200 IF (SLTDIS.LT.TOL.OR.ANG.LT.TOL) GO TO 500                      SLNT0150
C   SLANT DISTANCE AND ANGLE INPUT                                SLNT0160
C   VERDIS=SLTDIS*SIN(ANG*PIRAD)                                         SLNT0170
C   HORDIS=SQRT(SLTDIS**2-VERDIS**2)                                       SLNT0180
C   CONVERT TO 20 METER INCREMENTS                               SLNT0190
C 300 VERDIS=FLOAT(IFIX((VERDIS+TOL)*KMTOM/20.)) *20./KMTOM           SLNT0200
C   LIMIT ON VERTICAL HEIGHT IS 500 METERS                           SLNT0210
C   IF (VERDIS*KMTOM.GT.500.) VERDIS=.5                                SLNT0220
C   SD=SQRT(HORDIS**2+VERDIS**2)                                         SLNT0230
C   IF (SD/SLTDIS.GT.1.01.OR.SD/SLTDIS.LT..99) WRITE(I00UT,700)SLTDIS,SDSLNT0240
C   SLTDIS=SD                SLNT0250
C   FIND NBR OF 20 METER INCREMENTS                               SLNT0260
C   ITOP=IFIX((VERDIS+TOL)*KMTOM/20.)                                SLNT0270
C   IF (ITOP.LT.1) ITOP=1
C   VERDIS=FLOAT(ITOP)*20./KMTOM
C   EXTN=EXT55
C   BEGIN TRAPEZODIAL INTEGRATION FOR TAU ( OPTICAL DEPTH)          SLNT0280
C   TAU=EXTN/2,                SLNT0290
C   ALT=0.,0                 SLNT0300
C   NPTS=ITOP+1
C   IF (NPLT.EQ.1) WRITE (NPLOTU,352) NPTS
352  FORMAT (I5)
C   IF (NPLT.EQ.1) WRITE (NPLOTU,351) EXTN,ALT,WAVE
DO 400 I=1,ITOP
C   THESE FORMULAS ARE GOOD ONLY IN 20M INCREMENTS
C   IF (EXTN.GE.7.0.AND.WAVE.LT.2.0) EXTN=
1 10.**(.55+.72*ALOG10(EXTN))                                         SLNT0310
C   IF (EXTN.LT.7.0.AND.WAVE.LT.2.0) EXTN=
1 10.**(.1+.25*ALOG10(EXTN))                                         SLNT0320
C   IF (EXTN.GE.3.3.AND.(WAVE.GE.3.0.AND.WAVE.LT.5.0)) EXTN=
1 10.**(.55+.72*ALOG10(EXTN))                                         SLNT0330
C   IF (EXTN.LT.3.3.AND.(WAVE.GE.3.0.AND.WAVE.LT.5.0)) EXTN=
1 10.**(.3+.1.2*ALOG10(EXTN))                                         SLNT0340
C   IF (EXTN.GE.1.7.AND.(WAVE.GE.8.0.AND.WAVE.LT.12.0)) EXTN=
1 10.**(.5+.0.75*ALOG10(EXTN))                                         SLNT0350
C   IF (EXTN.LT.1.7.AND.(WAVE.GE.8.0.AND.WAVE.LT.12.0)) EXTN=
1 10.**(.4+.1.2*ALOG10(EXTN))                                         SLNT0360
C   ALT=FLOAT(I)*20,
C   IF (NPLT.EQ.1) WRITE (NPLOTU,351) EXTN,ALT
351  FORMAT (3(E10.4,1X))
400  TAU=TAU+EXTN
C   FINISH TRAP INTEGRATION
C   TAU=(TAU-EXTN/2.)*.02
C   FIND AVERAGE EXTINCTION VALUE FOR SLANT PATH.
C   AVEX55=TAU/VERDIS
C   RETURN
500  WRITE (IOOUT,600)
C   IERR=1
C   RETURN
C   600 FORMAT (1X,38$ERROR - IMPROPER INPUT FOR SUBROUTINE
1     34$SLANT: TRANSMISSION SET EQUAL TO 1)                            SLNT0550
700  FORMAT (1H,18$WARNING FROM SLANT/,1X,22$THE VERTICAL DISTANCE , SLNT0560
C   +38$EXCEEDS THE 500 METER UPPER LIMIT, OR /,1X,10$THIS NOT AN      SLNT0570
C   +29$INTEGER MULTIPLE OF 20 METERS,/,1X,28$SLANT DISTANCE CHANGED FRSLNT0580
C   +0M ,F7.4,4H TO ,F7.4,3H KM/)                                         SLNT0590
C   SLNT0600

```

SLNT0610

END

SUBROUTINE TURB(WAVE, IERR) TUP00010
 CALCULATES TURBULENCE INDUCED POINTING JITTER AND POWER SPECTRUM TUR00020
 FOR LASER TARGET DESIGNATOR AND TERMINAL HOMING SEEKER TUR00030
TUR00040
TUR00050
TUR00060
TUR00070
TUR00080
TUR00090
TUR00100
TUR00110
TUR00120
TUR00130
TUR00140
TUR00150
TUR00160
TUR00170
TUR00180
TUR00190
TUR00200
TUR00210
TUR00220
TUR00230
TUR00240
TUR00250
TUR00260
TUR00270
TUR00280
TUR00290
TUR00300
TUR00310
TUR00320
TUR00330
TUR00340
TUR00350
TUR00360
TUR00370
TUR00380
TUR00390
TUR00400
TUR00410
TUR00420
TUR00430
TUR00440
TUR00450
TUR00460
TUR00470
TUR00480
TUR00490
TUR00500
TUR00510
TUR00520
TUR00530
TUR00540
TUR00550
TUR00560
TUR00570
TUR00580
TUR00590
TUR00600
TUR00610
TUR00620
TUR00630
TUR00640
TUR00650
TUR00660
TUR00670
TUR00680
TUR00690
TUR00700

 CALCULATION FOR THE DESIGNATOR PATH ARE PERFORMED EACH TIME THIS ROUTINE IS REFERENCED. THE CALCULATIONS FOR THE SEEKER PATHS ARE PERFORMED ONLY WHEN THE DATA CARDS DVRY, CN2, OR V2 ARE INCLUDED IN THE INPUT SET.
 THE INPUT IS CARD ORDER INDEPENDENT, WITH THE SINGLE RESTRICTION THAT THE 'GO' CARD MUST BE THE LAST CARD OF THE DATA SET.
 THE DATA IS COMPLETELY IDENTIFIED BY THE ID IN COLUMNS 1-4 OF EACH CARD, FOLLOWED BY UP TO 7<REAL> FIELDS AS NEEDED, WITH 10 COLUMNS PER FIELD BEGINNING IN COL. 11.
 COMMENTS BELOW.

THE INPUT FORMAT IS A4,6X,7(E10.4)

THE INPUT OF A CN1, CN2, V1 AND V2 TYPE DATA CARD IS TERMINATED WHEN THE FIRST COEFFICIENT OF VALUE ZERO IS ENCOUNTERED. THE REMAINING DATA COEFFICIENTS ON THE CARD, IF ANY, ARE IGNORED.

 THE FOLLOWING ARE REQUIRED RECORDS FOR AT THE FIRST INPUT SET.

IDENT.	VARS.	DESCRIPTION
PARM		
	DIAM	LASER TARGET DESIGNATOR APERTURE DIAMETER IN METERS
	THET	LASER BEAMSPREAD ANGLE IN RADIANS
	TDOT	LASER BEAM SLUE RATE IN RADIANS/SECOND
	RANG	DISTANCE FROM DESIGNATOR TO TARGET IN METERS
	TIME	DURATION OF CALCULATIONS IN SECONDS
	M	NO. OF FREQUENCIES FOR WHICH POINTING JITTER POWER SPECTRUM IS TO BE CALCULATED, IF M=0, THEN DEFAULT TO M=512.
	***RANG AND RIV ARE RECALCULATED IF IGEOSW=1	
	***N1 AND N2 ARE CALCULATED WITHIN THE ROUTINE.	
	N1	NO OF SEGMENTS IN DESIGNATOR PATH
	N2	NO. OF SEGMENTS IN SEEKER TO TARGET PATH
	N1 AND N2 ARE SET EQUAL TO THE INDEX OF THE LAST NON-ZERO COEFFICIENT READ INTO CN1 AND CN2 RESPECTIVELY.	
CN1	IR2	STARTING INDEX VALUE OF CN1(I)
	(CN1(I), I=IR2, IR2+5)	VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT WITH ONE VALUE FOR EACH SEGMENT OF RANGE FROM LASER DESIGNATOR TO TARGET (METERS**(-2/3))
V1	IR2	STARTING INDEX VALUE OF V1(I)
	(V1(I), I=IR2, IR2+5)	SET OF VALUES OF CROSSWIND VELOCITY CORRESPONDING TO EACH SEGMENT OF RANGE FROM LASER DESIGNATOR TO TARGET (M/SEC)
DVRY	DIV	DIAMETER OF SEEKER APERTURE IN METERS.
	RIV	SEEKER RANGE TO TARGET IN METERS.
CN2	IR2	STARTING INDEX VALUE OF CN2(I)

```

C      (CN2(I), I=IR2, IR2+5)
C      CN2(I) VALUES OF REFRACTIVE INDEX STRUCTURE
C      CONSTANT FOR EACH SEGMENT OF RANGE FROM TARGET TO
C      SEEKER (METERS**(-2/3))          TUR00710
TUR00720
TUR00730
TUR00740
TUR00750
TUR00760
TUR00770
TUR00780
TUR00790
TUR00800
TUR00810
TUR00820
TUR00830
TUR00840
TUR00850
TUR00860
TUR00870
TUR00880
TUR00890
TUR00900
TUR00910
TUR00920
TUR00930
TUR00940
TUR00950
TUR00960
TUR00970
TUR00980
TUR00990
TUR00990
TUR01000
TUR01010
TUR01020
TUR01030
TUR01040
TUR01050
TUR01060
TUR01070
TUR01080
TUR01090
TUR01100
TUR01110
TUR01120
TUR01140
TUR01150
TUR01160
TUR01170
TUR01180
TUR01190
TUR01200
TUR01210
TUR01220
TUR01230
TUR01240
TUR01250
TUR01260
TUR01270
TUR01280
TUR01290
TUR01300
TUR01310
TUR01320
TUR01330
TUR01340
TUR01350
TUR01360
TUR01370
TUR01380
TUR01390
TUR01400

V2      IR2      STARTING INDEX VALUE OF V2(I)          TUR00710
V2      V2(I)    VALUES OF CROSSWIND VELOCITY FOR EACH SEGMENT
C      OF RANGE FROM TARGET TO SEEKER (M/SEC).          TUR00720
NPPS    NONE     PRINTS TABULAR VALUES OF POWER SPECTRUM VS FREQUENCY TUR00820
NPPJ    NONE     PRINTS TABULAR VALUES OF POINTING JITTER VS TIME   TUR00830
NPAL    NONE     PRINTS TABULAR VALUES OF BOTH POWER SPECTRUM (PS)   TUR00840
C      AND POINTING JITTER (PJ).          TUR00850
C      **IF THIS CARD IS MISSING NO TABLES WILL BE PRINTED (DEFAULT VALUE).
THE FOLLOWING IDENT RECORD IS ALWAYS REQUIRED.          TUR00860
TUR00870
TUR00880
TUR00890
GO      SIGNIFIES TO BEGIN EXECUTION FOR THIS DATA SET.
AFTER EXECUTION, ANOTHER SET OF INPUTS MAY BE          TUR00900
READ-IN FOLLOWED BY ANOTHER 'GO' CARD.
ANY VALUES ESTABLISHED FROM PREVIOUS INPUT SETS          TUR00910
TO THE ROUTINE ARE STILL IN EFFECT.  THUS DATA          TUR00920
SUCH AS FROM CARD PARM NEED NOT BE READ AGAIN IF          TUR00930
THERE ARE TO BE NO CHANGES IN THE DATA ASSOCIATED          TUR00940
WITH THAT IDENTIFIER.          TUR00950
TUR00960
TUR00970
TUR00980
TUR00990
*****+
++ CALLED PROGRAMS ++
DESUB
FALPH
FFT4
GAUSS
MEANVR
RAND
SPECT
SPREAD
THETO
C+++++
COMPLEX RAN
REAL LAMB
REAL INRC7, IR1, LABEL(11)
LOGICAL SETUP
COMMON /CONST/PI, PI2, PIRAD, TWOP, TORRMB, CDEGK
COMMON /IOUNIT/I0IN, I0OUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU
THIS IS A COMPLEX NUMBER EACH NUMBER TAKES TWO WORDS
COMMON /M05/RANK(2048)
COMMON /M01/FR(1025), CN1(20), V1(20), FD(20), RD(20)
COMMON /LOWEX/PS(1025), V2(20), RR(10)
DIMENSION CN2(20), PJCHAR(4), PSCHAR(4)
COMMON /GEOMET/PTS(15), IGEOSW
EXTERNAL DESUB, FALPH
DATA SETUP/.TRUE./
1 DATA PJCHAR, PSCHAR /4HPJ (,4HRAD*,4H*2/S,4HEC) ,4HPS (,4HRAD*,
1           4H*2/H,4H2) /
1 DATA LABEL/4HGO ,4H ,4HPARM, 4HCN1 ,4HV1 ,4HDVRV, 4HCN2 ,
1 4HV2 ,4HNPPS, 4HNPPJ, 4HNPAL/          TUR01270
TUR01280
TUR01290
TUR01300
TUR01310
TUR01320
TUR01330
TUR01340
TUR01350
TUR01360
TUR01370
TUR01380
TUR01390
TUR01400

C      NPRINT=0
C      LAMB=WAVE
C      SET THE SEED FOR THE RANDOM NUMBER GENERATOR
C*** NOTE, THIS SEED IS APPROPRIATE FOR THE RANDOM NUMBER GENERATOR
C*** USED AT THE ATMOSPHERIC SCIENCES LAB.  USERS AT OTHER
C*** INSTALLATIONS WILL NEED TO SUPPLY THEIR OWN RANDOM NUMBER
C*** GENERATOR.
IF (.NOT.SETUP) GO TO 100
VARX=735,34829
VARX=RAND(VARX)
SETUP=.FALSE.

```

```

100 IOPT=1
    READ(10IN,3000) IR1,(INR(I), I=1,7)
    IR2=IFIX(INR(1))
    IF(IR1.EQ.LABEL(1)) GO TO 180
    IF(IR1.EQ.LABEL(3)) GO TO 110
    IF(IR1.EQ.LABEL(4)) GO TO 120
    IF(IR1.EQ.LABEL(5)) GO TO 130
    IF(IR1.EQ.LABEL(6)) GO TO 140
    IF(IR1.EQ.LABEL(7)) GO TO 150
    IF(IR1.EQ.LABEL(8)) GO TO 160
C   PRINTING OPTIONS
    IF(IR1.EQ.LABEL(9)) NPRINT=1
    IF(IR1.EQ.LABEL(10)) NPRINT=2
    IF(IR1.EQ.LABEL(11)) NPRINT=3
    IF(NPRINT.EQ.1.OR.NPRINT.EQ.2.OR.NPRINT.EQ.3) GO TO 100
    WRITE(10OUT,3001) IR1,(INR(I), I=1,7)
    GOTO 100
C   110 DIAM = INR(1)
    THET = INR(2)
    TDOT = INR(3)
    RANG = INR(4)
    TIME = INR(5)
    M = IFIX(INR(6))
    IF(M.EQ.0) M = 512
    GOTO 100
C   120 DO 125 I=2,7
    IF(INR(I).NE.0.0) GOTO 121
    N1=IR2-1
    GOTO 100
121 CN1(IR2)=INR(I)
    IR2=IR2+1
    IF(IR2.GT.20) GOTO 126
125 CONTINUE
126 IR1=IR2-1
    N1=MAX0(N1,IFIX(IR1))
    GOTO 100
C   130 DO 131 I=2,7
    V1(IR2)=INR(I)
    IR2=IR2+1
    IF(IR2.GT.20) GOTO 100
131 CONTINUE
    GOTO 100
C   140 IOPT = 2
    DIV = INR(1)
    RIV = INR(2)
    GOTO 100
C   150 IOPT = 2
    DO 155 I=2,7
    IF(INR(I).NE.0.0) GOTO 151
    N2=IR2-1
    GOTO 100
151 CN2(IR2) = INR(I)
    IR2=IR2+1
    IF(IR2.GT.20) GOTO 156
155 CONTINUE
156 IR1=IR2-1
    N2=MAX0(N2,IFIX(IR1))
    GOTO 100
C   160 IOPT = 2
    DO 161 I=2,7
    V2(IR2) = INR(I)
    IR2=IR2+1
    IF(IR2.GT.20) GOTO 100
161 CONTINUE

```

TUR01410
TUR01420
TUR01430
TUR01440
TUR01450
TUR01460
TUR01470
TUR01480
TUR01490
TUR01500
TUR01510
TUR01520
TUR01530
TUR01540
TUR01550
TUR01560
TUR01570
TUR01580
TUR01590
TUR01600
TUR01610
TUR01620
TUR01630
TUR01640
TUR01650
TUR01660
TUR01670
TUR01680
TUR01690
TUR01700
TUR01710
TUR01720
TUR01730
TUR01740
TUR01750
TUR01760
TUR01770
TUR01780
TUR01790
TUR01800
TUR01810
TUR01820
TUR01830
TUR01840
TUR01850
TUR01860
TUR01870
TUR01880
TUR01890
TUR01900
TUR01910
TUR01920
TUR01930
TUR01940
TUR01950
TUR01960
TUR01970
TUR01980
TUR01990
TUR02000
TUR02010
TUR02020
TUR02030
TUR02040
TUR02050
TUR02060
TUR02070
TUR02080
TUR02090
TUR02100

```

GOTU 100                                     TUR02110
C 180 IF(ICEOSW.NE.1)GO TO 190                 TUR02120
      RANG=SQRT((PTS(7)-PTS(1))**2+(PTS(8)-PTS(2))**2+
      +(PTS(9)-PTS(3))**2)                         TUR02130
      DISKTM=1000.0                                TUR02140
      RANG=RANG*DISKTM                            TUR02150
      RIV=RANG*(PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+(PTS(6)-PTS(3))**2) TUR02160
      RIV=RIV*DISKTM                            TUR02170
190 CONTINUE                                    TUR02180
      IF (LAMB.LE.14.) GO TO 200                  TUR02190
      WRITE (IOOUT,2500) LAMB                      TUR02200
      IERR=1                                       TUR02210
      RETURN                                         TUR02220
200 WRITE (IOOUT,3100)                         TUR02230
      IF (IOPT.EQ.2) WRITE (IOOUT,3200)             TUR02240
      WRITE (IOOUT,3300) LAMB,DIAM,THET            TUR02250
C CHANGE WAVELENGTH TO METERS
      LAMB=LAMB/1.0E+6                            TUR02260
      IF (IOPT.EQ.2) WRITE (IOOUT,3400) D1V,RIV    TUR02270
      WRITE (IOOUT,3500) TDOT,RANG                TUR02280
      WRITE (IOOUT,3600) TIME                      TUR02290
      WRITE (IOOUT,3700) N1                        TUR02300
      IF (IOPT.EQ.2) WRITE (IOOUT,3800) N2        TUR02310
      WRITE (IOOUT,3900) M                         TUR02320
      WRITE (IOOUT,4000)                           TUR02330
      WRITE (IOOUT,4200)                           TUR02340
      DO 400 I=1,N1                               TUR02350
      WRITE (IOOUT,4300) I,CN1(I),V1(I)           TUR02360
400 CONTINUE                                    TUR02370
      IF (IOPT.EQ.1) GO TO 600                   TUR02380
      WRITE (IOOUT,4100)                           TUR02390
      WRITE (IOOUT,4200)                           TUR02400
      DO 500 I=1,N2                               TUR02410
      WRITE (IOOUT,4300) I,CN2(I),V2(I)           TUR02420
500 CONTINUE                                    TUR02430
C COMPUTATION OF TIME, FREQUENCY AND SPATIAL INCREMENTS
600 DELT=TIME/M                                TUR02440
      DELF=1./TIME                               TUR02450
      DELZ=RANG/FLOAT(N1)                         TUR02460
      IF (N2.NE.0) DEL1V=R1V/FLOAT(N2)             TUR02470
      MM=M+M                                     TUR02480
      M1=M+1                                     TUR02490
      MM1=MM+1                                   TUR02500
      MM2=MM+2                                   TUR02510
      MSG=SQRT(FLOAT(MM))                         TUR02520
      DO 700 I=2,M1                               TUR02530
      FR(I)=I-1)*DELF                            TUR02540
      FR(I)=I-1)*DELF                            TUR02550
      FR(I)=I-1)*DELF                            TUR02560
      FR(I)=I-1)*DELF                            TUR02570
      FR(I)=I-1)*DELF                            TUR02580
      FR(I)=I-1)*DELF                            TUR02590
      R2=DIAM/THET                             TUR02600
      R=RANG+R2                                 TUR02610
      DT=THET*RANG                            TUR02620
      D2=DIAM+DT                                TUR02630
C COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND
C NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM LASER
C DESIGNATOR TO TARGET
      ZI=0.0                                     TUR02640
      Z1=DELZ/2.                                 TUR02650
      ROT=0.0                                     TUR02660
      DEL=0.0                                     TUR02670
      DO 800 I=1,N1                               TUR02680
      ZI=ZI+Z1                                   TUR02690
      Z1=DELZ                                     TUR02700
      VEI=V1(I)+TDOT*(ZI-R2)                     TUR02710
      ROC(I)=16.71*DELZ*CN1(I)*(1.-ZI/RANG)**1.66667/(LAMB*LAMB)
      FC(I)=VEI/(PI*D2*ZI/R)                     TUR02720
      ROT=ROT+ROC(I)                            TUR02730
      DEL=DEL+DELZ*CH1(I)*(RANG-ZI)/RANG       TUR02740
      ROC(I)=ROC(I)**(-.6)                       TUR02750
      FC(I)=FC(I)**(-.6)                         TUR02760
      ROT=ROT+ROC(I)                            TUR02770
      DEL=DEL+DELZ*CH1(I)*(RANG-ZI)/RANG       TUR02780
      ROC(I)=ROC(I)**(-.6)                       TUR02790
      FC(I)=FC(I)**(-.6)                         TUR02800
800 CONTINUE

```

```

ROT=ROT**(-.6) TUR02810
CALL SPREAD(DIAM,ROT,LAMB,THET,RANG,DRO,THETDL,DDL,THET12,D12, TUF02820
      DTHET,D22) TUR02830
1 WRITE (IOOUT,4500) R2,R TUR02840
WRITE (IOOUT,4600) ROT,DRO,THET,DT,THETDL,DDL,THET12,D12,DTHET,D22 TUR02850
WRITE (IOOUT,4700) TUR02860
DO 1000 I=1,N1 TUR02870
1000 WRITE (IOOUT,4800) I,RO(I),FO(I) TUR02880
C COMPUTATION OF ANGLE OF ARRIVAL POWER SPECTRUM OF LASER DESIGNATOR TUR02890
F2=0. TUR02900
PS(1)=0. TUR02910
DO 1100 J=2,M1 TUR02920
F=FR(J) TUR02930
F1=0. TUR02940
CALL SPECT(F,D2,N1,F1,LAMB) TUR02950
PS(J)=F1 TUR02960
IF (IOP>EQ.1) PS(J)=F1*(D2/DIAM)**2 TUR02970
F2=F2+PS(J)*DELF TUR02980
1100 CONTINUE TUR02990
IF (IOP>EQ.1) WRITE (IOOUT,4400) F2 TUR03000
F2SQRT=SQRT(F2) TUR03010
AJITT=F2SQRT*RANG TUR03020
IF (IOP>EQ.1) WRITE (IOOUT,4900) F2SQRT,AJITT TUR03030
IF (IOP>EQ.1) GO TO 1500 TUR03040
C TUR03050
DTV=THET*R1V TUR03060
C COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND TUR03070
C NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM TUR03080
C TARGET TO SEEKER TUR03090
ZI=0. TUR03100
Z1=DEL1V/2. TUR03110
ROT=0.0 TUR03120
DO 1200 I=1,N2 TUR03130
ZI=ZI+Z1 TUR03140
Z1=DEL1V TUR03150
VEI=Y2(I)+TDOT*(R1V-ZI) TUR03160
RO(I)=16.71*DEL1V*CN2(I)*(ZI/R1V)**1.66667/(LAMB*LAMB) TUR03170
ROT=ROT+RO(I) TUR03180
RO(I)=RO(I)**(-.6) TUR03190
FO(I)=VEI/(PI*D1V*ZI/R1V) TUR03200
1200 CONTINUE TUR03210
ROT=ROT**(-.6) TUR03220
CALL SPREAD(D22,ROT,LAMB,THET,R1V,DRO,THETDL,DDL,THET12,D12,DTHET, TUR03230
      D22V) TUR03240
1 WRITE (IOOUT,4550) TUR03250
1 WRITE (IOOUT,4600) ROT,DRO,THET,DT,THETDL,DDL,THET12,D12,DTHET, TUR03260
1 D22V TUR03270
1 WRITE (IOOUT,4700) TUR03280
DO 1300 I=1,N2 TUR03290
1300 WRITE (IOOUT,4800) I,RO(I),FO(I) TUR03300
C COMPUTATION OF TURBULENCE INDUCED POINTING JITTER POWER SPECTRUM TUR03310
C FROM TARGET SPOT TO LASER SEEKER, COMPUTATION OF TOTAL POWER TUR03320
C SPECTRUM FROM LASER DESIGNATOR TO SEEKER AND POWER SPECTRUM VARIANCE TUR03330
CALL THETG(THETA0,FALPH,CN2,D1V,R1V,N2) TUR03340
F2=0. TUR03350
DO 1400 J=2,M1 TUR03360
F=FR(J) TUR03370
F1=0. TUR03380
CALL SPECT(F,D1V,N2,F1,LAMB) TUR03390
PS(J)=PS(J)+F1/(1.+(D2/(R1V*THETA0))**2) TUR03400
PS(J)=PS(J)*(D2/D1V)**2 TUR03410
F2=F2+PS(J)*DELF TUR03420
1400 CONTINUE TUR03430
WRITE (IOOUT,4400) F2 TUR03440
1500 IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5000) DELF TUR03450
FREQ=0.0 TUR03460
FINC=DELF*10.0 TUR03470
IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5600) TUR03480
IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PSCHAR TUR03490
K=1 TUR03500

```

```

1600 L=10
      IF (L.GT.M1) L=M1
      IF(NPRINT.EQ.1.OR.NPRINT.EQ.3)WRITE (IOOUT,2800)FREQ,(PS(J),J=K,L)
      FREQ=FREQ+FINC
      K=L+1
      L=L+9
      IF (K.LE.M1) GO TO 1600
      DO 2400 L=1,IOPT
      IF (L.EQ.1) WRITE (IOOUT,5100)
      IF (L.EQ.2) WRITE (IOOUT,5200)
C   GENERATION OF RANDOM SEQUENCE HAVING SAME POWER SPECTRUM VARIANCE
C   AS INDUCED BY TURBULENCE. ADD SYMMETRIC TERMS FOR NEGATIVE
C   FREQUENCIES. COMPUTE MEAN AND VARIANCE OF RANDOM ARRAY
      RANK(1)=(0.,0.)
      DO 1700 I=2,M1
      MMM=MM2-I
C   GENERATE RANDOM NUMBER WITH ROUTINE GAUSS
      NORMAL DISTRIBUTION
      MEAN = 0.0
      STANDARD DEVIATION = 1.0
      REAL PART = RANDOM NUMBER
      IMAG PART = 0
      RANK(I)=CMPLX(GAUSS(12,0.0,1.0),0.0)*SQRT(PS(I)/DELT)
      RANK(MMM)=RANK(I)
1700 CONTINUE
C   CALCULATE POWER OF 2 ,NPOW, FOR FFT4 SINCE ARRAY PASSED TO
C   FFT4 MUST HAVE SIZE THAT IS A POWER OF 2. (NOTE LN(2)=0.693147.)
      NPOW=IFIX ALOG(FLOAT(MM))/0.693147
      NMAX=2***NPOW
      IF (MM.EQ.NMAX) GO TO 1900
C   IF MM IS NOT A POWER OF 2 THEN RESET THE REST OF ARRAY RAN
      NPOW=NPOW+1
      ISTART=MM+1
      NMAX=2***NPOW
      DO 1800 I=ISTART,NMAX
      RANK(I)=(0.0,0.0)
1800 CONTINUE
C   COMPUTE AND WRITE MEAN AND VARIANCE OF RANDOM ARRAY
1900 WRITE (IOOUT,5300)
      CALL MEANVR(1,M)
C   FAST FOURIER TRANSFORM RANDOM ARRAY
C   CDC ROUTINE CALL
      CALL FFT(RAN,MM,+1)
      CALL FFT4(1.0,RAN,NPOW,NMAX)
      DO 2100 I=1,MM
      RANK(I)=RANK(I)/MSQ
2100 CONTINUE
C   COMPUTE AND WRITE MEAN AND VARIANCE OF TIME SEQUENCE.
      WRITE (IOOUT,5500)
      CALL MEANVR(M1,MM)
C   WRITE TRANSFORMED ARRAY VALUES CORRESPONDING TO TIME VALUES
C   OF POINTING JITTER FOR ONE DIRECTION.
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5400) DELT
      DTIME=DELT
      TINC=DELT*10.0
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5700)
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PJCHAR
      DO 2300 I1=M1,MM,10
      DO 2200 I2=1,10
      I3=I1+I2-1
      RR(I2)=REAL(RANK(I3))
2200 CONTINUE
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3)WRITE (IOOUT,2800)DTIME,
      1 (RR(I),I=1,10)
      DTIME=DTIME+TINC
2300 CONTINUE
2400 CONTINUE
C   2500 FORMAT (1X,100(1H*),/,13H WAVELENGTH (,F10.3,7H) GREAT,

```

1	40HER THAN 14 MICRONS: CONTROL RETURNED TO ,	TUR04210
2	21HMAIN FROM TURBULENCE.,/,1X,100<1H*>)	TUR04220
2700	FORMAT <5F16.6>	TUR04230
2800	FORMAT <E10.4,10E12.4>	TUR04240
3000	FORMAT <A4,6X,7E10.4>	TUR04250
3001	FORMAT <1H0,20X,45HTHE FOLLOWING ID-FIELD NOT RECOGNIZED BY TURB.,/,1X,A4,6X,7E10.4>	TUR04260
3100	FORMAT <1H1,36H CALCULATION OF POWER SPECTRUM AND 1 11HTURBULENCE 29HINDUCED POINTING JITTER OF A 2 23HLASER TARGET DESIGNATOR>	TUR04270
3200	FORMAT <1H+,10IX,10HAND SEEKER>	TUR04280
3300	FORMAT <1H0,41X,30HLASER WAVELENGTH <MICROMETERS>,8X, 1 F10.4,/,42X,26HDESIG. APERTURE DIAMETER < 2 7HMETERS>,5X,F10.6,/,42X,18HBEAMSPREAD ANGLE < 3 8RADIAN>,12X,F10.6>	TUR04290
3400	FORMAT <1H0,41X,33HSEEKER APERTURE DIAMETER <METERS>,5X, 1 F10.6,/,42X,36HRANGE FROM TARGET TO SEEKER <METERS>, 2 2X,F10.2>	TUR04300
3500	FORMAT <1H0,41X,24HBEAM SLUE RATE <RAD/SEC>,14X,F10.6, 1 //,42X,26HDESIGNATION RANGE <METERS>,12X, 2 F10.2>	TUR04310
3600	FORMAT <1H0,41X,26HDURATION OF TEST <SECONDS>,12X, 1 F10.4>	TUR04320
3700	FORMAT <1H0,41X,30HTOTAL DESIGNATOR PATH SEGMENTS,15X, 1 I3>	TUR04330
3800	FORMAT <1H0,41X,26HTOTAL SEEKER PATH SEGMENTS,19X,I3>	TUR04340
3900	FORMAT <1H0,41X,27HTOTAL FREQUENCIES FOR WHICH,/,42X, 1 22H POWER SPECTRUM IS TO,/,42X, 2 15H BE CALCULATED,27X,I4>	TUR04350
4000	FORMAT <1H0,/,26X,29H VALUES OF REFRACTIVE INDEX 1 37HSTRUCTURE CONSTANT AND WIND SPEED IN 2 15HDESIGNATOR PATH>	TUR04360
4100	FORMAT <1H0,/,26X,29H VALUES OF REFRACTIVE INDEX 1 37HSTRUCTURE CONSTANT AND WIND SPEED IN 2 11HSEEKER PATH>	TUR04370
4200	FORMAT <1H0,62X,5HCHN**2,9X,9HWINDSPEED,16X,/,42X, 1 11HSEGMENT NO.,5X,15H(METER**(-2/3)),3X, 2 11H(METER/SEC),//>	TUR04380
4300	FORMAT <1H,46X,I2,10X,E12.6,5X,F10.2>	TUR04390
4400	FORMAT <1H0,/,36H THE VARIANCE OF THE POWER SPECTRUM 1 3HIS,E12.4>	TUR04400
4500	FORMAT <1H1,/,56X,26HDESIGNATOR TO TARGET,/,38X, 1 //,24X,41HVIRTUAL POINT SOURCE TO APERTURE DISTANCE, 2 28X,F10.5,9H <METERS>,/,24X,22HDISTANCE FROM VIRTUAL 3 22HPOINT SOURCE TO TARGET,25X,F10.5,9H <METERS>>	TUR04410
4550	FORMAT <1H1,/,58X,16HTARGET TO SEEKER,//>	TUR04420
4600	FORMAT <1H0,23X,27HINTEGRATED COHERENCE LENGTH,42X,F10.6, 1 9H <METERS>,/,24X,36HDIAMETER/INTEGRATED COHERENCE LENGTHTUR04430	TUR04440
1	33X,F10.6,/,24X,31HTRANSMITTER-INDUCED BEAM SPREAD	TUR04450
2	13X,E12.5,10H <RADIAN>,3X,F10.6,9H <METERS>,	TUR04460
3	//,24X,31HDIFFRACTION-LIMITED BEAM SPREAD,13X,E12.5,	TUR04470
4	10H <RADIAN>,3X,F10.6,9H <METERS>,/,24X,	TUR04480
5	38HDIFFRACTION AND TURBULENCE BEAM SPREAD,6X,E12.5,	TUR04490
6	10H <RADIAN>,3X,F10.6,9H <METERS>,/,24X,6HTOTAL	TUR04500
7	19HEFFECTIVE BEAM SIZE,19X,E12.5,10H <RADIAN>,3X,F10.6,	TUR04510
8	9H <METERS>,//>	TUR04520
9		TUR04530
4700	FORMAT <1H0,34X,11HSEGMENT NO.,9X,16HCOHERENCE LENGTH, 1 5X,27H REFERENCE FREQUENCY(HERTZ),//>	TUR04540
4800	FORMAT <1H,38X,I2,10X,F16.6,10X,F16.6>	TUR04550
4900	FORMAT <1H0,19H RMS SPOT JITTER = ,E10.4,10H RAD, OR = 1 E10.4,7H METERS,/>	TUR04560
5000	FORMAT <1H1,1X,40X,31H CALCULATED POWER SPECTRUM VS. 1 9HFREQUENCY,/,47X,16H AT INTERVALS OF,F6.3,6H HERTZ>	TUR04570
5100	FORMAT <1H0,37H OUTPUT FOR DESIGNATOR TO TARGET PATH>	TUR04580
5200	FORMAT <1H0,33H OUTPUT FOR TARGET TO SEEKER PATH>	TUR04590
5300	FORMAT <1H0,35X,35H MEAN AND VARIANCE OF RANDOM ARRAY 1 //>	TUR04600
5400	FORMAT <1H1,49X,34HYVALUES OF POINTING JITTER VS. TIME / 1 52X,16H AT INTERVALS OF,F8.4,4H SEC>	TUR04610
5500	FORMAT <1H0,/,35X,28H MEAN AND VARIANCE OF TIME	TUR04620

```
i 8HSEQUENCE//>
5600 FORMAT (1H ,10H BEGINNING,/ ,10H FREQ (HZ))
5700 FORMAT (1H ,2X,5H TIME,/ ,2X,6H (SEC))
5800 FORMAT (1H+,T13,46(1H-),1H ,4R4,55(1H-),/ )
  RETURN
END
```

```
TUR04910
TUR04920
TUR04930
TUR04940
TUR04950
TUR04960
```

C FUNCTION DESUB(X,DRO)	DES00010
FUNCTION USED FOR INTEGRATION F(X)	DES00020
FACTR=3.44*(DRO*X)**1.666667*(1.-X**0.333333)	DES00030
IF(FACTR.LT.160.) GO TO 10	DES00040
DESUB=0.	DES00050
GO TO 20	DES00060
10 ARCCX=ATAN2(SQRT(1.-X**2),X)	DES00070
DESUB=X*((ARCCX-X*(1.-X**2)**.5)*EXP(-FACTR))	DES00080
20 RETURN	DES00090
END	DES00100

```

FUNCTION FALPH(XI1)
IF (XI1.GE.,.5623) GO TO 100
FALPH=10.66*( $(XI1)^{**2}$ )
GO TO 900
100 IF (XI1.GE.,0) GO TO 200
FALPH=4.025*XI1-.00659
GO TO 900
200 IF (XI1.GE.,1.778) GO TO 300
FALPH=1.8547*XI1+2.164
GO TO 900
300 IF (XI1.GE.,3.162) GO TO 400
FALPH=.8475*XI1+3.955
GO TO 900
400 IF (XI1.GE.,5.623) GO TO 500
FALPH=.391*XI1+5.397?
GO TO 900
500 IF (XI1.GE.,10.) GO TO 600
FALPH=.1814*XI1+6.578
GO TO 900
600 IF (XI1.GT.,31.62) GO TO 700
FALPH=.0534*XI1+7.95
GO TO 900
700 IF (XI1.GT.,1000.) GO TO 800
FALPH=7.8* $(XI1^{**.06})$ 
GO TO 900
800 FALPH=11.97
900 RETURN
END

```

FAL00010
FAL00020
FAL00030
FAL00040
FAL00050
FAL00060
FAL00070
FAL00080
FAL00090
FAL00100
FAL00110
FAL00120
FAL00130
FAL00140
FAL00150
FAL00160
FAL00170
FAL00180
FAL00190
FAL00200
FAL00210
FAL00220
FAL00230
FAL00240
FAL00250
FAL00260
FAL00270
FAL00280

```

SUBROUTINE FFT4(SIGN,X,NPOW,NMAX)          FFT0010
C                                     FFT0020
COOLEY-TUKEY METHOD OF FOURIER TRANSFORM   FFT0030
INCLUDES SINE COSINE COMPUTATION AND       FFT0040
REARRANGING DATA ACCORDING TO REVERSE BIT ADDRESSES FFT0050
SIGN = FOURIER DIRECTION TRANSFORM FLAG   FFT0060
    -1. FOR DIRECT TRANSFORM, TO COEFFICIENTS FROM SERIES FFT0070
    1. FOR INVERSE TRANSFORM, TO SERIES FROM COEFFICIENTS FFT0080
X      = LOC. OF FOURIER TRANSFORM BLOCK   FFT0090
NPOW = POWER OF 2 TO OBTAIN NMAX          FFT0100
NMAX = LENGTH OF BLOCK X                  FFT0110
COMPLEX X,CXCS,HOLD,XA                   FFT0120
DIMENSION CS(2),MSK(13)                  FFT0130
DIMENSION X(1)                          FFT0140
EQUIVALENCE (CXCS,CS)                  FFT0150
ZZ=6.283185306*SIGN/FLOAT(NMAX)        FFT0160
MSK(1)=NMAX/2                         FFT0170
DO 100 I=2,NPOW                      FFT0180
MSK(I)=MSK(I-1)/2                     FFT0190
100 CONTINUE                           FFT0200
NN=NMAX                               FFT0210
MM=2                                  FFT0220
C                                     FFT0230
LOOP OVER NPOW LAYERS                 FFT0240
DO 800 LAYER=1,NPOW                  FFT0250
NN=NN/2                                FFT0260
NW=0                                    FFT0270
DO 700 I=1,MM,2                      FFT0280
II=NN*I                            FFT0290
C                                     FFT0300
CXCS = CEXP(2*PI*NW*SIGN/NMAX)        FFT0310
W=FLOAT(NW)*ZZ                       FFT0320
CS(1)=COS(W)                         FFT0330
CS(2)=SIN(W)                         FFT0340
C                                     FFT0350
COMPUTE ELEMENTS FOR BOTH HALFS OF EACH BLOCK FFT0360
DO 200 J=1,NN                        FFT0370
II=II+1                                FFT0380
IJ=II-NN                               FFT0390
XA=CXCS*X(II)                         FFT0400
X(II)=X(IJ)-XA                        FFT0410
X(IJ)=X(II)+XA                        FFT0420
200 CONTINUE                           FFT0430
FFT0440
C                                     FFT0450
BUMP UP SERIES BY 2                  FFT0460
COMPUTE REVERSE ADDRESS             FFT0470
DO 400 LOC=2,NPOW                  FFT0480
LL=NW-MSK(LOC)                      FFT0490
IF (LL) 500,600,300                FFT0500
300 NW=LL                           FFT0510
400 CONTINUE                           FFT0520
500 NW=MSK(LOC)+NW                  FFT0530
GO TO 700                           FFT0540
600 NW=MSK(LOC+1)                  FFT0550
700 CONTINUE                           FFT0560
MM=MM*2                                FFT0570
C                                     FFT0580
800 CONTINUE                           FFT0590
DO FINAL REARRANGEMENT            FFT0600
NW=0                                    FFT0610
DO 1600 I=1,NMAX                  FFT0620
NW1=NW+1                             FFT0630
HOLD=X(NW1)                         FFT0640
IF (NW1-I) 1100,1000,900           FFT0650
900 X(NW1)=X(I)                      FFT0660
1000 X(I)=HOLD                      FFT0670
C                                     FFT0680
BUMP UP SERIES BY 1                  FFT0690
COMPUTE REVERSE ADDRESS             FFT0700
1100 DO 1300 LOC=1,NPOW           FFT0700
LL=NW-MSK(LOC)

```

```
1200 IF (LL) 1400,1500,1200
1300 NW=LL
1400 NW=MSK<LOC>+NW
      GO TO 1600
1500 NW=MSK<LOC+1>
1600 CONTINUE
      IF <SIGN> 1900,1900,1700
1700 PTS=NMAX
      DO 1800 I=1,NMAX
      X(I)=X(I)/PTS
1800 CONTINUE
1900 RETURN
END
```

```
FFT00710
FFT00720
FFT00730
FFT00740
FFT00750
FFT00760
FFT00770
FFT00780
FFT00790
FFT00800
FFT00810
FFT00820
FFT00830
FFT00840
```

```

C FUNCTION GAUSS(N,XBAR,SIGMA)
C GENERATE RANDOM NUMBERS WITH NORMAL DISTRIBUTION
C MEAN = XBAR
C STANDARD DEVIATION = SIGMA
C DATA NN /0/
C IF (NN.GT.0) GO TO 1
C NN=1
C=0.
1 CONTINUE
X=0.0
IF (C.EQ.0.0) C=735.34829
DO 100 J=1,N
C=RAND(C)
X=X+C
100 CONTINUE
XN=N
X=SQRT(12.0/XN)*(X-0.5*XN)
GAUSS=SIGMA*X+XBAR
RETURN
END

```

```

GAU00010
GAU00030
GAU00040
GAU00050
GAU00060
GAU00070
GAU00080
GAU00090
GAU00100
GAU00110
GAU00120
GAU00130
GAU00140
GAU00150
GAU00160
GAU00170
GAU00180
GAU00190
GAU00200
GAU00210

```

```

SUBROUTINE MEANVR(N1,N2)                                     MEA00010
C COMPUTE AND WRITE MEAN AND VARIANCE OF COMPLEX ARRAY      MEA00020
C OVER SOME RANGE OF THE ARRAY.                                MEA00030
C
C      ++ INPUT ++
C      N1 = STARTING INDEX OF RANGE                           MEA00040
C      N2 = ENDING INDEX OF RANGE                            MEA00050
C
C      ++ COMMON ++
C      RAN = COMPLEX ARRAY CONTAINING DATA                  MEA00060
C
C      ++ OUTPUT ++
C      MEAN1 = MEAN OF REAL PART                            MEA00070
C      MEAN2 = MEAN OF IMAGINARY PART                         MEA00080
C      VAR1 = VARIANCE OF REAL PART                          MEA00090
C
C      VAR2 = VARIANCE OF IMAGINARY PART                     MEA00100
C
C      COMPLEX RAN
C      REAL MEAN1,MEAN2                                     MEA00110
C      THIS IS A COMPLEX NUMBER EACH NUMBER TAKES TWO WORDS   MEA00120
C      COMMON /M05/RAN(2048)                                  MEA00130
C      COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUMEA00140
C      MEAN1=0;                                              MEA00150
C      MEAN2=0;                                              MEA00160
C      VAR1=0;                                              MEA00170
C      VAR2=0;                                              MEA00180
C      DO 100 I=N1,N2                                      MEA00190
C      MEAN1=REAL(RAN(I))+MEAN1                            MEA00200
C      MEAN2=AIMAG(RAN(I))+MEAN2                           MEA00210
C      VAR1=(REAL(RAN(I)))**2+VAR1                         MEA00220
C      VAR2=(AIMAG(RAN(I)))**2+VAR2                         MEA00230
C 100 CONTINUE                                           MEA00240
C      MEAN1=MEAN1/FLOAT(N2-N1)                            MEA00250
C      MEAN2=MEAN2/FLOAT(N2-N1)                            MEA00260
C      VAR1=VAR1/FLOAT(N2-N1)                            MEA00270
C      VAR2=VAR2/FLOAT(N2-N1)                            MEA00280
C      WRITE (I0OUT,200) MEAN1,MEAN2                      MEA00290
C      WRITE (I0OUT,300) VAR1,VAR2                         MEA00300
C      RETURN                                              MEA00310
C
C 200 FORMAT (35X,20H MEAN OF REAL PART =,E12.5,10H, MEAN OF  MEA00320
C ,11HIMAG PART =,E12.5)                                 MEA00330
C 300 FORMAT (35X,20H VAR. OF REAL PART =,E12.5,10H, VAR. OF  MEA00340
C ,11HIMAG PART =,E12.5)                                 MEA00350
C
C END                                                 MEA00360

```

```

C SUBROUTINE SPECT(F,D2,N,F1,LAMB)
C THIS ROUTINE GENERATES THE APPROXIMATE FUNCTION
C G<ALPHA> (F/F<SUB 0,I>)
C USED IN THE POWER SPECTRUM OF ANGLE-OF-ARRIVAL EQUATION
REAL LAMB
COMMON /MO1/FR(1025),CH(20),V1(20),FO(20),RO(20)
FACT=1.32E-2*(LAMB/D2)**2
F1=0.
DO 100 I=1,N
IF (F.LE..332*FO(I)) G=1
IF (F.GT..332*FO(I)) G=1.12-.361*F/FO(I)
IF (F.GE.3.10*FO(I)) G=0.
F1=F1+FACT*((D2/RO(I))**5/(F*F*FO(I)))**.33333*G
100 CONTINUE
RETURN
END

```

SPE00010
SPE00020
SPE00030
SPE00040
SPE00050
SPE00060
SPE00070
SPE00080
SPE00090
SPE00100
SPE00110
SPE00120
SPE00130
SPE00140
SPE00150
SPE00160

```

1 SUBROUTINE SPREAD(DIAM,ROT,WAVE,THET,RANG,DRO,THETDL,DDL,THET12,
      D12,DTHET,DTOT) SPR00010
      SPR00020
      SPR00030
      SPR00040
      SPR00050
      SPR00060
      SPR00070
      SPR00080
      SPR00090
      SPR000100
      SPR000110
      SPR000120
      SPR000130
      SPR000140
      SPR000150
      SPR000160
      SPR000170
      SPR000180
      SPR000190
      SPR000200
      SPR000210
      SPR000220
      SPR000230
      SPR000240
      SPR000250
C COMPUTATION OF BEAM SPREAD ANGLE DUE TO DIFFRACTION AND DIFFRACTION
C AND TURBULENCE AND SPOT DIAMETER ON TARGET (UR SEEKER).
C
C      DRO=DIAM/ROT
C      DO 1/2 SIMPSON RULE INTEGRATION
C      VARX=0.0
C      DELTAX=0.01
C      RDRO=DESUB(VARX,DRO)/2.0
C      DO 100 I=1,100
C      VARX=VARX+DELTAX
C      RDRO=RDRO+DESUB(VARX,DRO)
100   CONTINUE
C      RDRO=(RDRO-DESUB(VARX,DRO)/2.0)*DELTAX
C      RDRO=1.0/(SQRT(5.092958*(DRO)**2*RDRO))
C      THETDL=1.128*WAVE/DIAM
C      DDL=THETDL*RANG
C      THET12=THETDL*DRO*RDRO
C      D12=THET12*RANG
C      DTHET=SQRT(THET12**2+THET**2)
C      DTOT=DIAM+DTHET*RANG
C      RETURN
C      END

```

```

C      SUBROUTINE THETO(THETA0,FALPH,CN2,DIV,RIV,N2)          THE00010
C      ++ CALLED FUNCTIONS ++
C      FALPH           THE00020
C      DIMENSION CN2(20)          THE00030
C      COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU THE00040
C      DEL1V=R1V/FLOAT(N2)          THE00050
C      CALCULATE D1INF          THE00060
C      D1INF=0,          THE00070
C      S1=DEL1V/2,          THE00080
C      S=0,          THE00090
C      DIV3=(DIV)*(-.3333)          THE00100
C      DO 100 I=1,N2          THE00110
C      S=S+S1          THE00120
C      D1INF=DEL1V*CN2(I)*((S/RIV)**1.6667)+D1INF          THE00130
C      S1=DEL1V          THE00140
100   CONTINUE          THE00150
C      D1INF=0.5*11.97*DIV3*D1INF          THE00160
C      INITIAL ESTIMATE FOR THETA0          THE00170
C      WRITE (I0OUT,500) D1INF          THE00180
C      THETA0=1.E-4          THE00190
200   XI1=0,          THE00200
C      D1THE=0,          THE00210
C      S=0,          THE00220
C      S1=DEL1V/2,          THE00230
C      DO 300 I=1,N2          THE00240
C      S=S+S1          THE00250
C      XI1=THETA0*(RIV-S)/DIV          THE00260
C      D1THE=DEL1V*CN2(I)*((S/RIV)**1.6667)*FALPH(XI1)+D1THE          THE00270
C      S1=DEL1V          THE00280
300   CONTINUE          THE00290
C      D1THE=D1THE*D1V3          THE00300
C      IF (ABS((D1INF-D1THE)/D1INF),LT,.001) GO TO 400          THE00310
C      THETA0=THETA0*(1+.5*(D1INF-D1THE)/D1INF)          THE00320
C      WRITE (I0OUT,500) THETA0,D1THE          THE00330
C      GO TO 200          THE00340
400   CONTINUE          THE00350
C      WRITE (I0OUT,500) D1THE          THE00360
C      RETURN          THE00370
C      500 FORMAT (2E16.8)
END          THE00380
                                     THE00390
                                     THE00400
                                     THE00410

```

SUBROUTINE BASCAT(WAVE,EXCO,IERR)
 \$
 THIS VERSION OF BASCAT (20 SEP 81) DIFFERS FROM THE EOSAEL 80
 VERSION IN ITS INTERNAL PROGRAM STRUCTURE AND OUTPUT CAPABILITY.
 INPUT FORMATTING HAS NOT BEEN CHANGED. BRIEFLY, THE INTERNAL
 STRUCTURAL CHANGES CONSIST OF THE FOLLOWING :
 (A) SUBROUTINE THIT HAS BEEN ELIMINATED. THE FUNCTIONS WHICH
 IT ONCE PERFORMED HAVE BEEN CONSOLIDATED INTO SUBROUTINE
 START.
 (B) A NEW LIDAR BIASING ALGORITHM HAS BEEN INSERTED.
 (C) DIRECT BEAM (I.E., UNSCATTERED) COMPUTATIONS IN
 SUBROUTINE START HAVE BEEN REVISED.
 (D) SUBROUTINE CONV HAS BEEN MODIFIED SO THAT THE DIFFER-
 ENCE OF TWO NUMBERS RETAINS MORE SIGNIFICANT DIGITS.
 (E) NUMERICAL CHECKS FOR IMPROPER ARGUMENTS OF FUNCTIONS
 (DIVISIONS, SQUARE ROOTS, LOGARITHMS, ETC.) HAVE BEEN
 REVISED.
 (F) ARGUMENT LISTS OF A FEW COMMON BLOCKS HAVE BEEN CHANGED.
 (G) WRITE STATEMENTS FOR OUTPUTTING BASCAT RESULTS TO A USER-
 DEFINED PLOT FILE (NPLOTU) HAVE BEEN INCLUDED. THESE STATE-
 MENTS MUST BE UNCOMMENTED IN ORDER TO ACTIVATE THEM.
 \$
 THE BASCAT MODULE ALONE USES THE FOLLOWING SUBROUTINES:
 BKWD - CONTAINS BACKWARD SCATTERING ALGORITHM
 CONV - CONVOLVES IMPULSE RESPONSE WITH SQUARE PULSE
 ELM - DETERMINES BIASING DISTANCES
 FIND - DETERMINES INTERPOLATED PHASE FUNCTION VALUE
 FWRD - FIRST ORDER SCATTERING ALGORITHM
 GAS - DETERMINES MONTE CARLO SCATTERING ANGLES FOR TRAVERSSES
 GMAX - DETERMINES MAXIMUM OF AN INPUT ARRAY
 MATRX - GENERATES ROTATION MATRICES
 ROTAT - ROTATES VECTORS FROM ONE COORDINATE SYSTEM TO ANOTHER
 SMOOZ - DETERMINES START OF TRAILING ZEROS IN INPUT ARRAY
 START - INITIATES PHOTON TRAJECTORIES
 TRAVRS - MOVES PHOTONS BETWEEN SCATTERING POINTS AND FINDS OB-
 SERVED POWER CONTRIBUTIONS AT THOSE POINTS
 USCA - SELECTS RANDOM ANGLES WEIGHTED BY PHASE FUNCTION
 TWO SUBROUTINES SHARED BY BASCAT WITH OTHER EOSAEL 80 MODULES
 ARE THE FOLLOWING :
 PFUNC - SELECTS AND RENORMALIZES PHASE FUNCTION DATA FROM
 EOSAEL 80 DATA BASE
 RAND - RANDOM NUMBER GENERATOR (GENERATES UNIFORM DISTRIBUTION
 OF RANDOM NUMBERS BETWEEN 0 AND 1)
 ** NOTE** THE FOLLOWING ROUTINES UTILIZE THE RANDOM NUMBER GENERATOR
 WHICH IS INVOKED AS FUNCTION 'RAND(SEED)' :
 (A) BKWD - 3 OCCURRENCES OF FUNCTION RAND
 (B) FWRD - 2 OCCURRENCES
 (C) GAS - 3 OCCURRENCES
 (D) START - 2 OCCURRENCES
 (E) TRAVRS - 2 OCCURRENCES
 USERS OF OTHER (NON-MP) COMPUTER SYSTEMS MUST REPLACE
 FUNCTION RAND(SEED) WITH A UNIFORM RANDOM NUMBER GENERATOR
 WHICH WORKS FOR THEIR SYSTEMS. THE RANDOM NUMBER SEED IS
 INITIALIZED IN SUBROUTINE BASCAT. THIS AND ALL SUBSEQUENT
 VALUES OF THE RANDOM NUMBER SEED ARE PASSED VIA COMMON
 BLOCK 'RNDM'. SHOULD THE RANDOM NUMBER SEED USED HERE
 (735.34829) BE INAPPROPRIATE FOR THE USER'S SYSTEM, IT IS
 SUGGESTED THAT THE INITIALIZATION USED BELOW (SEED0=735.34829)
 BE CHANGED TO AN APPROPRIATE VALUE.
 SUBSTANTIAL MODIFICATIONS HAVE BEEN MADE TO THE FOLLOWING
 SUBROUTINES PRESENT IN EOSAEL 80 :
 CONV

BAS00001
 BAS00002
 BAS00003
 BAS00004
 BAS00005
 BAS00006
 BAS00007
 BAS00008
 BAS00009
 BAS00010
 BAS00011
 BAS00012
 BAS00013
 BAS00014
 BAS00015
 BAS00016
 BAS00017
 BAS00018
 BAS00019
 BAS00020
 BAS00021
 BAS00022
 BAS00023
 BAS00024
 BAS00025
 BAS00026
 BAS00027
 BAS00028
 BAS00029
 BAS00030
 BAS00031
 BAS00032
 BAS00033
 BAS00034
 BAS00035
 BAS00036
 BAS00037
 BAS00038
 BAS00039
 BAS00040
 BAS00041
 BAS00042
 BAS00043
 BAS00044
 BAS00045
 BAS00046
 BAS00047
 BAS00048
 BAS00049
 BAS00050
 BAS00051
 BAS00052
 BAS00053
 BAS00054
 BAS00055
 BAS00056
 BAS00057
 BAS00058
 BAS00059
 BAS00060
 BAS00061
 BAS00062
 BAS00063
 BAS00064
 BAS00065
 BAS00066
 BAS00067
 BAS00068
 BAS00069
 BAS00070

```

C     TRAVRS
C     START
BAS00710
BAS00720
BAS00730
BAS00740
BAS00750
BAS00760
BAS00770
BAS00780
BAS00790
BAS00800
BAS00810
BAS00820
BAS00830
BAS00840
BAS00850
BAS00860
BAS00870
BAS00880
BAS00890
BAS00900
BAS00910
BAS00920
BAS00930
BAS00940
BAS00950
BAS00960
BAS00970
BAS00980
BAS00990
BAS01000
BAS01010
BAS01020
BAS01030
BAS01040
BAS01050
BAS01060
BAS01070
BAS01080
BAS01090
BAS01100
BAS01110
BAS01120
BAS01130
BAS01140
BAS01150
BAS01160
BAS01170
BAS01180
BAS01190
BAS01200
BAS01210
BAS01220
BAS01230
BAS01240
BAS01250
BAS01260
BAS01270
BAS01280
BAS01290
BAS01300
BAS01310
BAS01320
BAS01330
BAS01340
BAS01350
BAS01360
BAS01370
BAS01380
BAS01390
BAS01400

***** THIS SUBROUTINE CALCULATES STEADY STATE AND TIME-DEPENDENT DIRECT
AND MULTIPLY SCATTERED POWER INTO A DETECTOR BY AN ELLIPSOIDAL
AEROSOL CLOUD WITH GROUND PLANE, FOR A LASER SOURCE. THE DETECTOR
AND SOURCE MAY HAVE ANY LOCATIONS, LOOK ANGLES, AND CONE OF VIEW/
BEAM SPREAD/WAVELENGTH. THE AEROSOL CLOUD MAY HAVE ANY ORIENTA-
TION, SIZE, AND SCATTERING PHASE FUNCTION (ARBITRARY NORMALIZA-
TION), IN A COORDINATE SYSTEM WITH ORIGIN AT THE CLOUD CENTER,
WITH Z-AXIS VERTICAL, X-AXIS EAST, AND Y-AXIS NORTH. THE GROUND
PLANE, ASSUMED AN ISOTROPIC REFLECTOR, MAY HAVE ANY ALBEDO, AND
MAY OR MAY NOT INTERSECT THE AEROSOL CLOUD.
*****  

*** INPUT DATA CARDS ARE READ IN AN ORDER-INDEPENDENT MANNER, WITH
*** A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH RECORD. DATA
*** ON EACH CARD IS READ IN UNDER THE FOLLOWING FORMAT :
*** (A4,1X,7(E9.4,1X)), NOTE THAT INTEGER VARIABLES IN THE PROGRAM
*** MUST BE INPUT AS REAL NUMBERS IN THIS INPUT SCHEME ... THEY ARE
*** LATER FIXED TO THE INTEGER TYPE.  

-----  

CARD IDENTIFIER : PART  

VARIABLES READ : N1,N2,ITIME  

N1=NUMBER OF PARTIAL OUTPUTS DESIRED, FOR A GIVEN RUN  

N2=NUMBER OF PHOTONS TO BE USED FOR EACH PARTIAL CALCULATION  

*** NOTE ** FOR CERTAIN DETECTOR CONDITIONS, AND WITH NORMAL BIASING,
AS MANY AS 100,000 PHOTONS MAY BE NEEDED TO OVERCOME LARGE
STATISTICAL FLUCTUATIONS IN FIRST ORDER SCATTERING RETURNS.
SUCH CONDITIONS ARE DEFINED BY THE FOLLOWING CHARACTERISTICS  

(A) THE DETECTOR IS IN A MONOSTATIC LIDAR CONFIGURATION.  

(B) THE DETECTOR IS WITHIN 10 METERS OF THE CLOUD (OR IS INSIDE
OF IT).  

(C) THE CLOUD IS NOT OPTICALLY THICK ALONG THE LOOK DIRECTION
(OPTICAL DEPTHS LESS THAN 10).  

*** IN ORDER TO ATTAIN MORE RAPID CONVERGENCE OF FIRST ORDER RETURN
POWER UNDER THE ABOVE CONDITIONS, A DIFFERENT BIASING SCHEME IS
USED. THE SPECIFIC CONDITIONS WHICH TRIGGER THIS ALTERNATE MODE
ARE THE FOLLOWING :  

(A) THE DOT PRODUCT OF THE SOURCE APERTURE SURFACE NORMAL AND
THE DETECTOR APERTURE SURFACE NORMAL IS GREATER THAN 0.99.  

(B) THE DISTANCE OF THE DETECTOR APERTURE FROM THE NEAREST
CLOUD SURFACE (AS SEEN ALONG THE DETECTOR NORMAL) IS LESS
THAN OR EQUAL TO 10 METERS (0.01 KM).  

(C) THE SEPARATION OF SOURCE AND DETECTOR APERTURE CENTERS IS
LESS THAN OR EQUAL TO 10 TIMES THE DETECTOR APERTURE
RADIUS.
```

*** THE MAIN DIFFERENCES WHICH WILL BE OBSERVED BETWEEN THE NORMAL AND
 ALTERNATE BIASING MODES ARE THE FOLLOWING :
 (A) THE FIRST ORDER RETURN POWER IN THE EARLIEST TIME BOX WILL
 BE EXTREMELY STABLE IN THE ALTERNATE MODE.
 (B) ONLY THE EARLIEST FIRST ORDER SCATTERING RESULTS WILL BE
 STRONGLY AFFECTED.
 (C) LATER FIRST ORDER RETURNS AND ALL HIGHER ORDER RETURNS WILL
 HAVE NEGLIGIBLY POORER CONVERGENCE.
 *** IN ORDER TO OBSERVE CONVERGENCE OF THE FIRST ORDER STEADY-STATE
 RETURN POWER TOWARD A STABLE VALUE, IT IS SUGGESTED THAT THE USER
 UTILIZE THE PARTIAL OUTPUT OPTION. AS AN EXAMPLE, IF 10,000
 PHOTONS ARE REQUIRED, SET N1=10 AND N2=1,000. THIS SELECTION WILL
 RUN $10 \times 1,000 = 10,000$ PHOTONS AND WILL OUTPUT RETURN POWER
 RESULTS AFTER EACH BATCH OF 1,000 PHOTONS.
 ITIME=OVERALL RUN NUMBER FOR THIS SET OF PARAMETERS. FOR EXAMPLE,
 ITIME=1 MEANS THE FIRST RUN, ITIME=3 MEANS THAT THE RESULTS
 OF THE TWO PREVIOUS RUNS WILL BE COMBINED WITH THIS 3RD RUN.

CARD IDENTIFIER : SORC
 VARIABLES READ : XS(1),XS(2),XS(3),THES,PHIS,ASMM
 (XS(K),K=1,3)=SOURCE XYZ COORDINATES(KM)
 THES,PHIS=(POLAR,AZIMUTHAL)ANGLES(DEG) OF SOURCE BEAM AXIS
 ASMM=RADIUS OF SOURCE APERTURE(CM). THE SOURCE BEAM SPREAD ANGLE
 'THSP' IS TAKEN BY THE PROGRAM AS THE DIFFRACTION LIMIT FOR
 THIS APERTURE. IF YOU SET ASMM=0, THE PROGRAM PUTS THSP=0.
 IF YOUR SOURCE PHOTONS WOULD NOT INTERSECT THE CLOUD OR THE
 GROUND, THE SUBROUTINE NOTIFIES YOU, AND RETURNS.

CARD IDENTIFIER : DTR
 VARIABLES READ : XD(1),XD(2),XD(3),THED,PHID,THEV,ACM
 (XD(K),K=1,3)=DETECTOR XYZ COORDINATES(KM)
 THED,PHID=(POLAR,AZIMUTHAL)ANGLES(DEG) OF DETECTOR LOOK AXIS
 THEV=DETECTOR CONE OF VIEW HALF-ANGLE(DEG)
 ACM=RADIUS OF DETECTOR DISK(CM)
 IF YOUR DETECTOR POINTS SKYWARD, OR IF NEITHER YOUR SOURCE NOR
 YOUR DETECTOR LOOK INTO THE CLOUD, THE SUBROUTINE NOTIFIES YOU,
 AND RETURNS.

CARD IDENTIFIER : CLDS
 VARIABLES READ : A(1),A(2),A(3),THE,PHE,PSE,ISO
 (A(K),K=1,3)=ELLIPSOIDAL CLOUD PRINCIPAL HALF-AXES(KM)
 THE,PHE,PSE=ELLIPSOID EULER ANGLES(DEG), WHERE 'PHE' IS THE FIRST
 ROTATION, ABOUT THE Z-AXIS, 'THE' IS THE NEXT, ABOUT THE NEW
 Y-AXIS, 'PSE' IS THE LAST, ABOUT THE NEW Z-AXIS.
 ISO=AEROSOL IDENTIFICATION NUMBER, TO COMPARE WITH THE 'ID'
 PARAMETER.

CARD IDENTIFIER : GRND
 VARIABLES READ : ZG,ALBG
 ZG=Z-COORDINATE OF GROUND PLANE(KM)
 ALBG=GROUNd PLANE REFLECTIVITY 0. <ALBG>.
 IF YOUR ZG IS SUCH THAT THE GROUND PLANE IS ENTIRELY ABOVE THE
 CLOUD, THE SUBROUTINE INFORMS YOU, AND RETURNS. IF YOUR ZG IS
 NEGATIVE, AND SO LARGE THAT NO GROUND REFLECTIONS WILL RETURN TO
 THE DETECTOR WITHIN THE TIME LIMIT SET BY THE SUBROUTINE, THE SUB-
 ROUTINE SETS ALBG=0. IF YOUR SOURCE IS BELOW THE GROUND PLANE,
 THE SUBROUTINE PUTS THE SOURCE ON THE GROUND PLANE, AT YOUR XY
 COORDINATES, AND NOTIFIES YOU.

CARD IDENTIFIER : PULS
 VARIABLES READ : TPU(1),TPU(2),...,TPU(7)
 (TPU(J),J=1,7)=SOURCE PULSE DURATIONS(USEC)
 YOU CAN INPUT AS MANY AS SIX DIFFERENT PULSE LENGTHS. THE LAST
 ENTRY MUST BE BLANK(ZERO). THE SUBROUTINE CONVOLUTES THE MONTE
 CARLO PROBABILITY PER UNIT TIME DATA WITH EACH OF THESE SQUARE
 PULSES, AND, FOR EACH PULSE, WRITES THE TIME-DEPENDENT POWER TO
 THE DETECTOR, FOR UNIT SOURCE PULSE POWER, FOR EACH SIGNIFICANT
 ORDER OF MULTIPLE SCATTERING, AND FOR THE TOTAL OF ALL ORDERS.

```

CARD IDENTIFIER : GO
VARIABLES READ : NONE
RUN TERMINATION CARD (MUST BE LAST CARD READ).
IERR=0

C*** THE FOLLOWING STATEMENT MAY NEED TO BE CHANGED OR ELIMINATED
FOR USE ON NON-HP1000 COMPUTERS.

SEED0=735.34829
NDIM=100
ND1=NDIM-1
DTAUM=0.2
GAMMA=EXCO
SINGWV=WAVE

C** INPUT DATA INITIALIZATIONS.
IF(IZERO.NE.0) GO TO 600
N1=0
N2=0
ITIME=0
IS0=0
DO 599 LL=1,3
XS(LL)=0.
XD(LL)=0.
AY(LL)=0.
TPU(LL)=0.
599 TPU(LL+3)=0.
TPU(7)=0.
THES=0.
PHIS=0.
ASMM=0.
THED=0.
PHID=0.
THEV=0.
ACM=0.
THE=0.
PHE=0.
PSE=0.
ZG=0.
ALBG=0.
LLMAX=65
IZERO=1
600 CONTINUE

C*** READ BASCAT DATA SET RECORDS UNDER CARD-INDEPENDENT FORMAT
DO 700 K=1,7
READ(IOIN,610) IA,IA1,(DAT(I),I=1,7)
610 FORMAT(2A2,1X,7E9.4,1X))
DO 615 JJ=1,8
IF(IA.NE.IA1(JJ)) GO TO 615
INOPT=JJ
IF(INOPT.EQ.7) GO TO 701
GO TO 620
615 CONTINUE
IF((K.EQ.7).AND.(JJ.EQ.8)) GO TO 697
GO TO 695
620 CONTINUE
GO TO (621,622,623,624,625,626),INOPT
621 N1=IFIX(DAT(1))
N2=IFIX(DAT(2))
ITIME=IFIX(DAT(3))
GO TO 700
622 XS(1)=DAT(1)
XS(2)=DAT(2)
XS(3)=DAT(3)
THES=DAT(4)
PHIS=DAT(5)

```

```

ASMM=DAT(6)                                BAS02810
GO TO 700                                  BAS02820
623  XD(1)=DAT(1)                            BAS02830
     XD(2)=DAT(2)                            BAS02840
     XD(3)=DAT(3)                            BAS02850
     THED=DAT(4)                            BAS02860
     PHID=DAT(5)                            BAS02870
     THEV=DAT(6)                            BAS02880
     ACM=DAT(7)                            BAS02890
     GO TO 700                                BAS02900
624  AC(1)=DAT(1)                            BAS02910
     AC(2)=DAT(2)                            BAS02920
     AC(3)=DAT(3)                            BAS02930
     THE=DAT(4)                            BAS02940
     PHE=DAT(5)                            BAS02950
     PSE=DAT(6)                            BAS02960
     ISO=IFIX(DAT(7))                      BAS02970
     GO TO 700                                BAS02980
625  ZG=DAT(1)                                BAS02990
     ALBG=DAT(2)                            BAS03000
     GO TO 700                                BAS03010
626  CONTINUE                                 BAS03020
     DO 627  NN=1,7                          BAS03030
627  TPU(NN)=DAT(NN)                        BAS03040
     GO TO 700                                BAS03050
C
C***  ERROR RETURNS
C
695  CONTINUE                                 BAS03060
     WRITE(I00UT,696)                         BAS03070
696  FORMAT(1H0,20X,89H***BASCAT ERROR*** INPUT CARD DETECTED WHICH DOES
     +S NOT MATCH ANY CORRECT INPUT IDENTIFIERS /)
     IERR=1                                    BAS03100
     GO TO 777                                BAS03130
697  CONTINUE                                 BAS03140
     WRITE(I00UT,698)                         BAS03150
698  FORMAT(1H0,20X,66H***BASCAT ERROR*** TOO MANY INPUT CARDS OR GO SEE
     +NTINEL NOT PRESENT /)
700  CONTINUE                                 BAS03170
701  CONTINUE                                 BAS03190
C
C***  GEOMETRICAL OPTION DATA TRANSFER
C
     IF(IGEOSW.NE.1) GO TO 111                BAS03200
     DO 110  I=1,3                           BAS03230
     XS(I)=PTS(I+6)-PTS(I+12)                 BAS03240
110   XD(I)=PTS(I+3)-PTS(I+12)                 BAS03250
     111  CONTINUE                               BAS03260
C
C***  GENERATE INTERPOLATED, RENORMALIZED PHASE FUNCTION
C
     CALL PFUNC(ISO)                          BAS03300
     IF(GAMMA.EQ.0.0) GAMMA=BE(1)            BAS03310
     ALBEDO=ALBED(1)                          BAS03320
     REWIND IPHFUN                            BAS03330
     ALB(1)=ALBEDO                            BAS03340
     ALB(2)=ALBG                               BAS03350
C
C***  DETERMINE POWER OF 2 (KMAX) CORRESPONDING TO NUMBER OF PHASE
C***  FUNCTION VALUES PRESENT.
C
     LMAX=LLMAX                             BAS03360
     LMM1=LMAX-1                            BAS03370
     KMAX=IFIX ALOG(FLOAT(LMM1))/0.693147)    BAS03380
C
C***  CC = SPEED OF LIGHT (KM/MICROSECOND)
C
     CC=0.3                                  BAS03400
     THV=PIRAD*THEV                          BAS03410
     IF(THV.LE.1.E-30) THV=0.                  BAS03420
                                         BAS03430
                                         BAS03440
                                         BAS03450
                                         BAS03460
                                         BAS03470
                                         BAS03480
                                         BAS03490
                                         BAS03500

```

```

        UV=COS(THV)
C*** SET RANDOM NUMBER SEED
C      SEED=(2*ITIME-1)*SEED0
      AKM=ACM*1.E-5
      AKSQ=AKM**2
C*** SET MINIMUM DISTANCE ALIM (AND ITS SQUARE BLIM) SEPARATING
C*** THE DETECTOR DISK AND A PHOTON SCATTERING POINT.
C      ALIM=AKM*10.
      BLIM=ALIM**2
      FAC=THV*AKSQ/8.
      BT=.61*WAVE*1.E-3
      THSP=0.
C*** DETERMINE DIFFRACTION-LIMITED SOURCE BEAMSPREAD
C      IF(ASMM.GT.0.)THSP=BT/ASMM
C*** GENERATE ROTATION MATRIX RE( ) FOR CONVERSION FROM STANDARD
C*** FRAME OF REFERENCE TO CLOUD FRAME OF REFERENCE
C      CALL MATRX(THE,PHE,SS,R)
      AT=PIRAD*PSE
      IF(AT.LE.1.E-30) AT=0.
      DO 1 I=1,3
      DO 1 J=1,3
1     RS(I,J)=0.
      RS(1,1)=COS(AT)
      RS(1,2)=SIN(AT)
      RS(2,1)=-RS(1,2)
      RS(2,2)=RS(1,1)
      RS(3,3)=1.
      DO 2 I=1,3
      DO 2 J=1,3
      RE(I,J)=0.
      DO 2 K=1,3
2     RE(I,J)=RE(I,J)+RS(I,K)*R(K,J)
C*** GENERATE ROTATION MATRIX RS( ) FOR CONVERSION FROM SOURCE
C*** CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE
C      CALL MATRX(THES,PHIS,SS,RS)
C*** ECHO INPUT PARAMETERS
C      WRITE (IOOUT,6800)
      WRITE (IOOUT,4800)
      WRITE (IOOUT,4900)
      WRITE (IOOUT,5000)
      WRITE (IOOUT,4800)
      WRITE (IOOUT,4700)
      WRITE (IOOUT,5100)
      IF (ISO.EQ.0) WRITE (IOOUT,5200)
      IF (ISO.EQ.1) WRITE (IOOUT,5201)
      IF (ISO.EQ.2) WRITE (IOOUT,5202)
      IF (ISO.EQ.3) WRITE (IOOUT,5203)
      IF (ISO.EQ.4) WRITE (IOOUT,5204)
      IF (ISO.EQ.5) WRITE (IOOUT,5205)
      IF (ISO.EQ.6) WRITE (IOOUT,5206)
      IF (ISO.EQ.7) WRITE (IOOUT,5207)
      IF (ISO.EQ.8) WRITE (IOOUT,5208)
      IF (ISO.EQ.9) WRITE (IOOUT,5209)
      IF (ISO.EQ.10) WRITE (IOOUT,5210)
      IF (ISO.EQ.11) WRITE (IOOUT,5211)
      IF (ISO.EQ.12) WRITE (IOOUT,5212)
      WRITE (IOOUT,5600) WAVE,ALBEDO
      WRITE (IOOUT,5700) GAMMA

```

```

      WRITE( IOOUT, 5601 )                                BAS04210
      WRITE( IOOUT, 5602 )                                BAS04220
      WRITE( IOOUT, 5603 )                                BAS04230
      WRITE( IOOUT, 5701 )                                BAS04240
      WRITE( IOOUT, 5702 )(XS(K),K=1,3)                  BAS04250
      WRITE( IOOUT, 5704 )THES                           BAS04260
      WRITE( IOOUT, 5705 )PHIS                           BAS04270
      WRITE( IOOUT, 5703 )ASMM                           BAS04280
      WRITE( IOOUT, 5706 )THSP                           BAS04290
      WRITE( IOOUT, 5900 )                                BAS04300
      WRITE( IOOUT, 6000 ) THEV                           BAS04310
      WRITE( IOOUT, 6100 ) ACM                            BAS04320
      WRITE( IOOUT, 6201 )(XD(K),K=1,3)                  BAS04330
      WRITE( IOOUT, 6400 ) THED                           BAS04340
      WRITE( IOOUT, 6500 ) PHID                           BAS04350
      WRITE( IOOUT, 6501 )                                BAS04360
      WRITE( IOOUT, 6502 )ZG,ALBG                         BAS04370
      DO 10 K=1,3                                     BAS04380
 10  ASQ(K)=A(K)**2                                 BAS04390
      DO 12 K=1,3                                     BAS04400
 12  Y(K)=A(K)*2.*GAMMA                           BAS04410
C*** DETERMINE LARGEST OPTICAL DEPTH PRESENT IN AEROSOL CLOUD   BAS04420
C CALL GMAX(3,TAU)                                    BAS04430
C*** SET TIME AND DISTANCE INCREMENTS AND LIMITS          BAS04440
C
      NTMAX=50                                         BAS04450
      D=TAU/GAMMA                                     BAS04460
      DELD=5.*D/NTMAX                                BAS04470
      DELT=DELD/CC                                     BAS04480
      DMAX=5.5*D                                       BAS04490
C*** BEGIN CLOUD SUBBLOCK                          BAS04500
C#####
C THE FOLLOWING BLOCK OF WRITE STATEMENTS MUST BE UNCOMMENTED    BAS04510
C IN ORDER TO OUTPUT DATA TO A USER-DEFINED PLOT FILE (NPLOTU).  BAS04520
C
C THE OUTPUT QUANTITIES IN THIS BLOCK ARE THE FOLLOWING :          BAS04530
      WAVE = WAVELENGTH (MICROMETERS)                         BAS04540
      ISO = AEROSOL TYPE (VALID RANGE, 0-12)                 BAS04550
      TAU = OPTICAL DEPTH ALONG LONGEST AXIS OF CLOUD ELLIPSOID  BAS04560
      N1 = NUMBER OF PARTIAL RUNS WITHIN THIS BASCAT RUN        BAS04570
      XS( ) = SOURCE XYZ POSITION ARRAY (KILOMETERS)           BAS04580
      THES = SOURCE VECTOR POLAR ANGLE (DEGREES)                BAS04590
      PHIS = SOURCE VECTOR AZIMUTHAL ANGLE (DEGREES)             BAS04600
      ASMM = SOURCE APERTURE RADIUS (MILLIMETERS)               BAS04610
      THSP = HALF-ANGLE OF SOURCE DIFFRACTION CONE (RADIAN)       BAS04620
      XD( ) = DETECTOR XYZ POSITION ARRAY (KILOMETERS)           BAS04630
      THED = DETECTOR VECTOR POLAR ANGLE (DEGREES)              BAS04640
      PHID = DETECTOR VECTOR AZIMUTHAL ANGLE (DEGREES)            BAS04650
      ACM = DETECTOR APERTURE RADIUS (CENTIMETERS)              BAS04660
      THEY = HALF-ANGLE OF DETECTOR FIELD OF VIEW (DEGREES)     BAS04670
      AC( ) = CLOUD ELLIPSOID PRINCIPAL HALF-AXIS ARRAY (KILOMETERS)  BAS04680
      ALB(1)= SINGLE-SCATTERING ALBEDO OF CLOUD AEROSOL         BAS04690
      ALB(2)= ALBEDO OF GROUND PLANE                           BAS04700
C
      WRITE( NPLOTU, 9111 ) WAVE,ISO,TAU,N1                  BAS04710
C9111  FORMAT(E9.4,1X,I2,1X,E9.4,1X,2(I2,1X))          BAS04720
C      WRITE( NPLOTU, 9222 )(XS(1),XS(2),XS(3),THES,PHIS,ASMM,THSP)  BAS04730
C      WRITE( NPLOTU, 9222 )(XD(1),XD(2),XD(3),THED,PHID,ACM,THEY)  BAS04740
C9222  FORMAT(12(E9.4,1X))                                BAS04750
C      WRITE( NPLOTU, 9222 ) AC(1),AC(2),AC(3),ALB(1),ALB(2)    BAS04760
C#####
C      WRITE( IOOUT, 5900 )                                BAS04770
      WRITE( IOOUT, 7000 )(A(K),K=1,3)                      BAS04780
      WRITE( IOOUT, 7100 )THE,PHE,PSE                         BAS04790
C
      WRITE( IOOUT, 5900 )                                BAS04800
      WRITE( IOOUT, 7000 )(A(K),K=1,3)                      BAS04810
      WRITE( IOOUT, 7100 )THE,PHE,PSE                         BAS04820
      WRITE( IOOUT, 5900 )                                BAS04830
      WRITE( IOOUT, 7000 )(A(K),K=1,3)                      BAS04840
      WRITE( IOOUT, 7100 )THE,PHE,PSE                         BAS04850
      WRITE( IOOUT, 5900 )                                BAS04860
      WRITE( IOOUT, 7000 )(A(K),K=1,3)                      BAS04870
      WRITE( IOOUT, 7100 )THE,PHE,PSE                         BAS04880
      WRITE( IOOUT, 5900 )                                BAS04890
      WRITE( IOOUT, 7100 )THE,PHE,PSE                         BAS04900

```

```

      WRITE(1000,7200)(Y(K),K=1,3)                                BAS04910
C*** GENERATE ROTATION MATRIX R( ) FOR CONVERSION FROM DETECTOR    BAS04920
C*** CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE        BAS04930
C   CALL MATRX(THED,PHID,SD,R)                                       BAS04940
C*** SET LIMIT NSCAM ON HIGHEST SIGNIFICANT SCATTERING ORDER (NSCAM-1) BAS04950
C
      NSCAM=3.0*ALBEDO*TAU+2.0                                     BAS05000
      IF(NSCAM.GT.10) NSCAM=10                                      BAS05010
      IF(NSCAM.LT.3)NSCAM=3                                         BAS05020
      NSCA1=NSCAM-1                                                 BAS05030
      AT=0.                                                       BAS05040
      DO 4 K=1,3                                                 BAS05050
 4   AT=AT+RE(K,3)**2/ASQ(K)                                       BAS05060
      BT=1.-AT*ZG                                              BAS05070
      IF(BT.LE.0.)GO TO 4002                                     BAS05080
      AT=ZG+D                                              BAS05090
      IF(AT.GT.0.)GO TO 3                                         BAS05100
      ZG=-D                                              BAS05110
      WRITE(1000,9004)ZG                                         BAS05120
 3   CONTINUE                                                 BAS05130
      IF(XS(3).GE.ZG)GO TO 16                                     BAS05140
      XS(3)=ZG                                              BAS05150
      WRITE(1000,9003)XS(3)                                       BAS05160
 16  CONTINUE                                                 BAS05170
C*** DETERMINE DISTANCES ELD(1), ELD(2) FROM DETECTOR TO NEAREST AND    BAS05180
C*** FARTHEST CLOUD BOUNDARIES ALONG DETECTOR AXIS                  BAS05190
C   CALL ELM(XD,SD,ELD)                                         BAS05200
C*** DETERMINE DISTANCES EL(1), EL(2) FROM SOURCE TO NEAREST AND       BAS05210
C*** FARTHEST CLOUD BOUNDARIES ALONG SOURCE AXIS                   BAS05220
C   CALL ELM(XS,SS,EL)                                         BAS05230
C*** PERFORM GEOMETRICAL ERROR CHECKS                               BAS05240
C
      IF((EL(2).LE.0.),AND,(ELD(2).LE.0.))GO TO 4003          BAS05250
      IF((EL(2).GT.0.),AND,(ELD(2).GT.0.))GO TO 6             BAS05260
      IF(EL(2).LE.0.)GO TO 7                                     BAS05270
      IF(ALB(2).LE.0.) GO TO 4004                           BAS05280
      IF(SD(3).GE.0.)GO TO 4001                           BAS05290
      AT=EL(1)
      CT=(ZG-XD(3))/SD(3)                                     BAS05300
      GO TO 9                                                 BAS05310
 7   CONTINUE                                                 BAS05320
      IF(ALB(2).LE.0.) GO TO 4005                           BAS05330
      IF(SS(3).GT.0.)GO TO 4000                           BAS05340
      AT=(ZG-XS(3))/SS(3)                                     BAS05350
      GO TO 8                                                 BAS05360
 6   AT=EL(1)                                               BAS05370
C*** DETERMINE MINIMUM POSSIBLE TRAVERSE DISTANCE ELMIN           BAS05380
C
 8   CT=ELD(1)                                              BAS05390
 9   BT=0.                                                 BAS05400
 10  DO 11 K=1,3                                           BAS05410
      X(K)=XS(K)+AT*SS(K)                                     BAS05420
      Y(K)=XD(K)+CT*SD(K)                                     BAS05430
 11  BT=BT+(X(K)-Y(K))**2                                    BAS05440
      IF(BT.LE.1.E-30) BT=0.                                     BAS05450
      ELMIN=SQRT(BT)+AT+CT                                     BAS05460
      IF(ELMIN.LT.ALIM) ELMIN=ALIM                           BAS05470
C*** DETERMINE VIRTUAL SOURCE POINT XVC >                      BAS05480
C
      DSA=0.                                                 BAS05490

```

```

IF(ASMM.GT.0.)DSA=ASMM*1.E-6/TAN(THSP)
DO 305 K=1,3
305 XV(K)=XS(K)-DSA*SS(K)

C*** DETERMINE WHETHER NEAR-CLOUD LIDAR BIASING IS NEEDED.
C
DOTSD=0.
DOTXD=0.
DO 47 KL=1,3
47 DOTSD=DOTSD+(XD(KL)-XS(KL))**2
    DOTSD=DOTSD+SS(KL)*SD(KL)
    IF(DOTXD.GT.BLIM) DOTSD=0.
    REWIND KSTOR

C*** START PHOTON LOOPS
C
DO 1000 II=1,N1

C*** CHECK TO SEE IF THIS IS THE FIRST RUN WITH NO PRIOR RESULTS USED
C
IF((II.EQ.1).AND.(ITIME.EQ.1))GO TO 1303

C*** RELOAD WITH RESULTS FROM PREVIOUS PARTIAL RUN
C
DO 1301 NSCA=1,NSCAM
READ(KSTOR)ENC(NSCA)
DO 1301 II=1,NTMAX
READ(KSTOR)EN(NSCA,II)
1301 CONTINUE
REWIND KSTOR
GO TO 1302
1302 CONTINUE
1303 CONTINUE

C*** INITIALIZATION FOR FIRST RUN
C
DO 1300 NSCA=1,NSCAM
ENC(NSCA)=0.
DO 1300 NT=1,NTMAX
1300 EN(NSCA,NT)=0.0
1302 CONTINUE

C*** LOAD EXPECTED CUMULATIVE NUMBER OF PHOTONS
C
DO 1304 NS=1,NSCAM
1304 ENC(NS)=ENC(NS)+N2

C*** DETERMINE VECTOR FROM VIRTUAL SOURCE POINT TO DETECTOR
C
DO 555 K=1,3
555 Y(K)=XD(K)-XV(K)

C*** START INNER PHOTON LOOP
C
DO 900 I2=1,N2

C*** INITIALIZE PHOTON DIRECTION AND CALCULATE DIRECT BEAM
C*** CONTRIBUTIONS (IF ANY)
C
CALL START(SS)
IF(STH.GT.0.)GO TO 150
GO TO 900
150 NSCA=1

C*** START MULTIPLE SCATTERING LOOP
C
I2FLG=0
ICOND=0
1700 NSCA=NSCA+1

C*** NEAR-CLOUD LIDAR BIASING IS ACTIVE WHEN ICOND=1. 10 PERCENT OF ALLBAS06300

```

```

C***** FIRST ORDER PHOTONS USE AN ALTERNATE MODE OF BIASING IN THIS      BAS06310
C***** SITUATION. THE REMAINING 90 PERCENT ARE NORMALLY BIASED IN ALL      BAS06320
C***** ORDERS.                                BAS06330
C                                         BAS06340
C                                         IF((NSCA.NE.2).OR.(DOTSD.LE.0.99).OR.(ELD(1).GT.0.01))GO TO 707  BAS06350
CICOND=1                                         BAS06360
C                                         IF(FLOAT(I2).LE.(FLOAT(N2)/10.))I2FLG=1  BAS06370
C 707 CONTINUE                                         BAS06380
C                                         BAS06390
C***** MOVE PHOTON TO NEXT EVENT POINT VIA BIASED TRAVERSE AND DETERMINE  BAS06400
C***** POWER CONTRIBUTIONS                                         BAS06410
C                                         BAS06420
C                                         CALL TRAVRS(JTYPE,I2FLG,ICOND)  BAS06430
C                                         IF((NSCA.EQ.NSCA1).OR.(STH.LE.0.))GO TO 900  BAS06440
C***** DETERMINE SCATTERING DIRECTION FOR NEXT TRAVERSE  BAS06450
C                                         BAS06460
C                                         CALL GASC(JTYPE)  BAS06470
C***** DETERMINE BIASING DIRECTIONS FOR NEXT TRAVERSE  BAS06480
C                                         BAS06490
C                                         CALL ELM(XA,SA,EL)  BAS06500
C                                         IF(EL(2).LE.EL(1))GO TO 900  BAS06510
C                                         GO TO 1700  BAS06520
C 900 CONTINUE                                         BAS06530
C                                         BAS06540
C***** END MULTIPLE SCATTERING LOOP, BEGIN CONVOLUTION BLOCK  BAS06550
C                                         BAS06560
C***** WRITE PARTIAL RUN RESULTS INTO STORAGE FILE FOR USE BY NEXT RUN  BAS06570
C                                         BAS06580
C                                         DO 2201 NSCA=1,NSCAM  BAS06590
C                                         WRITE(KSTOR)ENC(NSCA)  BAS06600
C                                         DO 2201 II=1,NTMAX  BAS06610
C                                         WRITE(KSTOR)ENC(NSCA,II)  BAS06620
C 2201 CONTINUE                                         BAS06630
C                                         REWIND KSTOR  BAS06640
C                                         BAS06650
C***** OUTPUT PARTIAL RUN RESULTS FOR STEADY STATE POWER  BAS06660
C                                         BAS06670
C                                         WRITE(100UT,7400)  BAS06680
C                                         WRITE(100UT,7500)  BAS06690
C                                         WRITE(100UT,7600)  BAS06700
C                                         PTOT=0.  BAS06710
C ##### THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO  BAS06720
C A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.  BAS06730
C                                         BAS06740
C                                         NSCAM= 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING  BAS06750
C                                         BAS06760
C                                         WRITE(NPLOTU,9333) NSCAM  BAS06770
C ##### THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO  BAS06780
C A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.  BAS06790
C                                         BAS06800
C                                         DO 2110 NS=1,NSCAM  BAS06810
C                                         PE=0.  BAS06820
C                                         NS1=NS-1  BAS06830
C                                         DO 2115 NT=1,NTMAX  BAS06840
C                                         PE=PE+ENC(NS,NT)  BAS06850
C                                         PE=PE/ENC(NS)  BAS06860
C                                         PTOT=PTOT+PE  BAS06870
C ##### THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO  BAS06880
C A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.  BAS06890
C                                         BAS06900
C                                         NS1 = ORDER OF SCATTERING  BAS06910
C                                         PE = OBSERVED STEADY STATE POWER FOR THIS ORDER  BAS06920
C                                         ENC() = TOTAL NUMBER OF PHOTONS COUNTED FOR THIS ORDER  BAS06930
C                                         WRITE(NPLOTU,9333) NS1,PE,ENC(NS)  BAS06940
C ##### THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO  BAS06950
C A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.  BAS06960
C                                         BAS06970
C                                         WRITE(100UT,7700)NS1,PE,ENC(NS)  BAS06980
C 2110 CONTINUE                                         BAS06990
C                                         BAS07000

```

```

THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO
A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.

PTOT = OBSERVED TOTAL STEADY STATE POWER (ALL ORDERS)

C9333 FORMAT(12,1X,2(E9.4,1X))
C      WRITE(NPLOTU, 9222) PTOT
C#####
C      WRITE(I0OUT,7701)PTOT
C      DO 2500 NSCA=1,NSCAM
C
C*** DETERMINE INDEX OF TIME OF LAST NONZERO POWER VALUE FOR EACH
C*** ORDER
C
C      CALL SMOOZ(NSCA,NO)
NM(NSCA)=1+NO
I0R(NSCA)=NSCA-1
2500 Y(NSCA)=NM(NSCA)
C
C*** DETERMINE LATEST TIME INDEX OF NONZERO POWER FOR ALL ORDERS
C
CALL GMAX(NSCAM,YMAX)
NMA=YMAX
JP=1
2900 JP=JP+1
AT=TPU(JP)
IF (AT.LE.0.0) GO TO 3000
GO TO 2900
3000 JPMAX=JP-1
WRITE (I0OUT,7400)
WRITE (I0OUT,7900) JPMAX
C
C*** BEGIN PULSE LOOP
C
DO 3800 JP=1,JPMAX
WRITE (I0OUT,7400)
TP=TPU(JP)
AT=(NMA+4)*DELT
IF(TP.GT.AT)TP=AT
IF(TP.LE.DELT)TP=DELT+1.E-3
NP=1.001+TP/DELT
NMAX=NMA+NP
IF(NMAX.GT.NDIM)NMAX=NDIM
NP=NMAX-NMA
TP=(NP-1)*DELT
TMAX=(NMAX-1)*DELT
WRITE (I0OUT,8100) JP,TP
WRITE (I0OUT,8200) JP,TMAX
IF(TP.LE.0.)GO TO 3999
DO 3400 NSCA=1,NSCAM
NMS=NM(NSCA)
NMS1=NMS-1

C*** NORMALIZE RETURN POWER BY DIVIDING CUMULATIVE POWER BY CUMULATIVE
C*** NUMBER OF PHOTONS
C
DO 3200 N=1,NMS1
3200 EN(NSCA,N)=EN(NSCA,N)/EN(NSCA)

C*** PERFORM SQUARE SOURCE PULSE CONVOLUTION WITH PROBABILITIES
C*** PER UNIT TIME
C
CALL CONV(NP,NMS,NMAX,NSCA)
3400 CONTINUE
DO 3500 N=1,NMAX
YN=0.,0.
DO 3500 NSCA=1,NSCAM
YN=Y(N)+EN(NSCA,N)
XN=-DELT
DO 3600 N=1,NMAX

```

```

      XN=XN+DELT          BAS07710
3600  X(N)=XN          BAS07720
C***  OUTPUT TIME-DEPENDENT RESULTS          BAS07730
C   WRITE( IOOUT,8300)          BAS07740
C   WRITE( IOOUT,8400)          BAS07750
C   WRITE( IOOUT,8500)X(I0R(NS),NS=1,NSCAM)          BAS07760
C   WRITE( IOOUT,8501)          BAS07770
C### THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO          BAS07780
C A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.          BAS07790
C
C   NMAX = NUMBER OF TIME BOXES USED FOR TIME-DEPENDENT DATA          BAS07800
C   JP = INDEX OF INPUT PULSE          BAS07810
C   NSCAM = 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING          BAS07820
C   X( ) = ARRAY OF TIME VALUES FOR EACH TIME BOX          BAS07830
C   Y( ) = ARRAY OF OBSERVED TIME-DEPENDENT TOTAL POWER          BAS07840
C   EN(L, ) = ARRAY OF OBSERVED TIME-DEPENDENT POWER FOR ORDER L+1          BAS07850
C
C   IF(I1.LT.N1) GO TO 9777          BAS07860
C   WRITE( NPLOTU,9444) NMAX,JP,NSCAM          BAS07870
C9444 FORMAT(3(I3,1X))          BAS07930
C   WRITE( NPLOTU,9222)X(LLL),LLL=1,NMAX          BAS07940
C   WRITE( NPLOTU,9222)Y(LLL),LLL=1,NMAX          BAS07950
C   DO 9666 LLL=1,NSCAM          BAS07960
C   WRITE( NPLOTU,9222)EN(LLL,LLX),LLX=1,NMAX          BAS07970
C9666 CONTINUE          BAS07980
C9777 CONTINUE          BAS07990
C### DO 3700 N=1,NMAX          BAS08000
C   WRITE( IOOUT,8600) X(N),Y(N),EN(NSCA,N),NSCA=1,NSCAM          BAS08010
3700 CONTINUE          BAS08020
C*** RELOAD PARTIAL RUN RESULTS FOR CONVOLUTION WITH NEXT PULSE          BAS08030
C
C   DO 3701 NS=1,NSCAM          BAS08040
C   READ(KSTOR)EN(NS)          BAS08050
C   DO 3701 II=1,NTMAX          BAS08060
C   READ(KSTOR)EN(NS,II)          BAS08080
3701 CONTINUE          BAS08090
C   REWIND KSTOR          BAS08100
3800 CONTINUE          BAS08110
C*** END PULSE LOOP          BAS08120
C   1000 CONTINUE          BAS08130
C*** END OUTER PHOTON LOOP?          BAS08140
C   777 RETURN          BAS08150
C*** ERROR RETURN MESSAGES          BAS08160
C
C   3999 CONTINUE          BAS08170
C   WRITE( IOOUT,8700)NTMAX          BAS08180
C   RETURN          BAS08190
4000 CONTINUE          BAS08200
C   WRITE( IOOUT,8800)          BAS08210
C   RETURN          BAS08220
4001 CONTINUE          BAS08230
C   WRITE( IOOUT,9000)          BAS08240
C   RETURN          BAS08250
4002 CONTINUE          BAS08260
C   WRITE( IOOUT,9001)          BAS08270
C   RETURN          BAS08280
4003 CONTINUE          BAS08290
C   WRITE( IOOUT,9002)          BAS08300
C   RETURN          BAS08310
4004 CONTINUE          BAS08320
C   4004 CONTINUE          BAS08330
C   4004 CONTINUE          BAS08340
C   4004 CONTINUE          BAS08350
C   4004 CONTINUE          BAS08360
C   4004 CONTINUE          BAS08370
C   4004 CONTINUE          BAS08380
C   4004 CONTINUE          BAS08390
C   4004 CONTINUE          BAS08400

```

```

        WRITE(1005)
        RETURN
4005  CONTINUE
        WRITE(1006)
        RETURN
C*****FORMATS
4200  FORMAT(7F10.5)                                BAS0841
4400  FORMAT(7I10)                                 BAS0842
4500  FORMAT(I10,2F10.5,I10)                         BAS0843
4600  FORMAT(5E15.10)                               BAS0844
4700  FORMAT(1H,34X,51(1H*))                         BAS0845
4800  FORMAT(1H,34X,1H*,49X,1H*)                   BAS0846
4900  FORMAT(1H,34X,1H*,9X,31HMONTE CARLO MULTIPLE SCATTERING,9X,1H*) BAS0847
5000  FORMAT(1H,34X,1H*,16X,18HAEROSOL SCATTERING,15X,1H*) BAS0848
5100  FORMAT(1H0,48X,23HPARAMETERS FOR THIS RUN)    BAS0849
5200  FORMAT(1H0,46X,28HUSER SUPPLIED PHASE FUNCTION) BAS0850
5201  FORMAT(1H0,43X,34HMARITIME ARCTIC, VIS=0.1 TO 2.0 KM) BAS0851
5202  FORMAT(1H0,47X,26HMARITIME POLAR, VIS=0.2 KM)   BAS0852
5203  FORMAT(1H0,47X,26HMARITIME POLAR, VIS=2.0 KM)   BAS0853
5204  FORMAT(1H0,42X,36HCONTINENTAL POLAR, VIS=0.2 TO 2.5 KM) BAS0854
5205  FORMAT(1H0,52X,16HWHITE PHOSPHORUS)             BAS0855
5206  FORMAT(1H0,52X,16HHEXACHLOROETHANE)            BAS0856
5207  FORMAT(1H0,57X,7HFOG OIL)                      BAS0857
5208  FORMAT(1H0,45X,31HDUST <MODERATE AEROSOL LOADING>) BAS0858
5209  FORMAT(1H0,46X,28HDUST <HEAVY AEROSOL LOADING>) BAS0859
5210  FORMAT(1H0,43X,34HMARITIME MODEL B, VIS=5 KM, RH=95%) BAS0860
5211  FORMAT(1H0,43X,35HMARITIME MODEL B, VIS=10 KM, RH=90%) BAS0861
5212  FORMAT(1H0,43X,35HMARITIME MODEL B, VIS=30 KM, RH=50%) BAS0862
5500  FORMAT(1H0,46X,27HUSER SUPPLIED AEROSOL MODEL)  BAS0863
5600  FORMAT(1H,36X,11HWAVELENGTH=,F6.3,16H MICROMETERS,7HALBEDO=,F5.3) BAS0864
      ,                                             BAS0865
5601  FORMAT(1H0,47X,25HELLIPSOIDAL AEROSOL CLOUD)  BAS0866
5602  FORMAT(1H,41X,36HCOORDINATE ORIGIN AT CENTER OF CLOUD)  BAS0867
5603  FORMAT(1H,38X,42HZ-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH) BAS0868
5700  FORMAT(1H,36X,31HAEROSOL EXTINCTION COEFFICIENT=,E10.4,7H KM**-1) BAS0869
      ,                                             BAS0870
5701  FORMAT(1H0,51X,17HSOURCE PARAMETERS)           BAS0871
5702  FORMAT(1H,36X,27HSOURCE XYZ COORDINATES(KM)=,3(F8.4,1X)) BAS0872
5703  FORMAT(1H,36X,27HSOURCE APERTURE RADIUS(MM)=,F7.3)  BAS0873
5704  FORMAT(1H,36X,26HSOURCE AXIS POLAR ANGLE =,F7.3,8H DEGREES) BAS0874
5705  FORMAT(1H,36X,26HSOURCE AXIS AZIMUTH ANGLE=,F7.3,8H DEGREES) BAS0875
5706  FORMAT(1H,36X,26HSOURCE BEAM SPREAD ANGLE =,E10.4,8H RADIANS) BAS0876
5900  FORMAT(1H0,50X,19HDETECTOR PARAMETERS)        BAS0877
6000  FORMAT(1H,35X,29H CONE OF VIEW HALF-ANGLE =,F7.3,8H DEGREES) BAS0878
      ,                                             BAS0879
6100  FORMAT(1H,35X,29H DETECTOR APERTURE RADIUS =,F7.3,3H CM) BAS0880
      ,                                             BAS0881
6201  FORMAT(1H,36X,29HDETECTOR XYZ COORDINATES(KM)=,3(F8.4,1X)) BAS0882
6400  FORMAT(1H,36X,28HDETECTOR AXIS POLAR ANGLE =,F7.3,8H DEGREES) BAS0883
      ,                                             BAS0884
6500  FORMAT(1H,36X,28HDETECTOR AXIS AZIMUTH ANGLE=,F7.3,8H DEGREES) BAS0885
      ,                                             BAS0886
6501  FORMAT(1H0,47X,23HGROUND PLANE PARAMETERS,/,40X,38HISOTROPIC REFLECTION FROM GROUND PLANE) BAS0887
6502  FORMAT(1H,36X,33HGROUND PLANE Z-COORDINATE ZG(KM)=,F7.3,/,37X,33HGROUND PLANE ALBEDO, ALBG, =,F7.3) BAS0888
      ,                                             BAS0889
6800  FORMAT(1H0,34X,51(1H*))                      BAS0890
6900  FORMAT(1H0,51X,16HCLOUD PARAMETERS)          BAS0891
7000  FORMAT(1H,31X,43HELLIPSOID PRINCIPAL XYZ HALF-AXES(KM) =,3(F8.4,1X)) BAS0892
      ,                                             BAS0893
7100  FORMAT(1H,31X,43HEULER ANGLES THE,PHE,PSE OF ELLIPSOID(DEG) =,3(F8.4,1X)) BAS0894
      ,                                             BAS0895
7200  FORMAT(1H,31X,43HOPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES =,3(F8.4,1X)) BAS0896
      ,                                             BAS0897
7400  FORMAT(1H0,100X)                            BAS0898
7500  FORMAT(1H0,32X,53HSTEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE * POWER) BAS0899
      ,                                             BAS0900
7600  FORMAT(1H0,37X,5HORDER,3X,18HSTEADY STATE POWER,3X,17HNUMBER OF PHOTONS) BAS0901
      ,                                             BAS0902
7700  FORMAT(1H,39X,I2,8X,E10.5,8X,E12.6)          BAS0903
      ,                                             BAS0904

```

```

7701 FORMAT (1H ,37X,5HTOTAL,7X,E10.5)          BAS09110
7900 FORMAT (1H0,32X,23HPOWER INTO DETECTOR FOR,I2,6H PULSE    BAS09120
1      23H<8> OF DIFFERENT LENGTH)               BAS09130
8100 FORMAT (1H0,37X,12HPULSE NUMBER,I2,12H HAS LENGTH ,      BAS09140
1      E10.4,13H MICROSECONDS)                   BAS09150
8200 FORMAT (1H ,23X,34HDETECTOR RESPONSE CUTOFF TIME FOR    BAS09160
1      12HPULSE NUMBER,I2,4H IS ,E10.4,              BAS09170
2      13H MICROSECONDS)                         BAS09180
8300 FORMAT (1H0,25X,68HDETECTOR RESPONSE, POWER AS A FUNCTION OF TIME,BAS09190
* FOR UNIT PULSE POWER)                         BAS09200
8400 FORMAT (1H0,55X,21HPOWER FROM EACH ORDER/14X,5HTOTAL)    BAS09210
8500 FORMAT (1H ,3X,4HTIME,6X,5HPOWER,2X,10<5X,12,4X>)     BAS09220
8501 FORMAT(1H ,130<1H->)                      BAS09230
8600 FORMAT (12<E10.4,1X>)                      BAS09240
8700 FORMAT(1H0,6HNTMAX=,I3,80H SHOULD BE DECREASED TO 46. IT IS TOO LABAS09250
*ERGE TO ALLOW CONVOLUTION WITH YOUR PULSE)           BAS09260
8800 FORMAT(1H0,97HYOUR INCIDENT PHOTONS NEVER INTERSECT THE CLOUD OR T   BAS09270
*HE GROUND. CHECK YOUR INPUT SOURCE PARAMETERS)        BAS09280
9000 FORMAT(1H0,72HTHE DETECTOR LOOKS ABOVE THE CLOUD. CHECK YOUR INPUTBAS09290
* DETECTOR PARAMETERS)                            BAS09300
9001 FORMAT(1H0,65HYOUR GROUND PLANE IS ENTIRELY ABOVE YOUR CLOUD. CHECBAS09310
*K YOUR INPUTS)                                BAS09320
9002 FORMAT(1H0,76HNEITHER YOUR SOURCE NOR YOUR DETECTOR LOOK INTO THE BAS09330
*CLOUD. CHECK YOUR INPUTS)                       BAS09340
9003 FORMAT(1H0,71HYOUR SOURCE WAS UNDERGROUND. IT HAS BEEN PUT AT THE BAS09350
*GROUND, WITH XS<3>=,F6.3,2HKM)                 BAS09360
9004 FORMAT(1H0,114HYOUR GROUND PLANE WAS TOO FAR AWAY FROM THE CLOUD T   BAS09370
*O PRODUCE GROUND REFLECTIONS WITHIN THE MAX TIME DELAY ALLOWED.,/,BAS09380
*34H THE GROUND PLANE WAS MOVED TO ZG=,F6.3,2HKM)       BAS09390
9005 FORMAT(1H0,91HGROUND PLANE WAS ABSENT AND DETECTOR DOES NOT LOOK ABAS09400
*T CLOUD. CHECK YOUR DETECTOR PARAMETERS)            BAS09410
9006 FORMAT(1H0,79HGROUND PLANE WAS ABSENT AND SOURCE DOES NOT ILLUMINABAS09420
*TE CLOUD. CHECK YOUR INPUTS)                      BAS09430
END

```

```

SUBROUTINE BKWD(JTYPE)
COMMON /CONST/PI,PI2,PIRAD,TWOP,I,PIRAD,TORRMB,CDEGK
COMMON/RNDM/ SEED
COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),EN(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASBOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+B(16),SINGWV,PFC(65),LLMAX
COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
*ALB(2),ZG,DMAX
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
COMMON/FGEL/XA(3),D,NSCA
DIMENSION SBC(3),SBA(3)
NSCAP=NSCA+1
BKW000110
BKW000120
BKW000130
BKW000140
BKW000150
BKW000160
BKW000170
BKW000180
BKW000190
BKW000200
BKW000210
BKW000220
BKW000230
BKW000240
BKW000250
BKW000260
BKW000270
BKW000280
BKW000290
BKW000300
BKW000310
BKW000320
BKW000330
BKW000340
BKW000350
BKW000360
BKW000370
BKW000380
BKW000390
BKW000400
BKW000410
BKW000420
BKW000430
BKW000440
BKW000450
BKW000460
BKW000470
BKW000480
BKW000490
BKW000500
BKW000510
BKW000520
BKW000530
BKW000540
BKW000550
BKW000560
BKW000570
BKW000580
BKW000590
BKW000600
BKW000610
BKW000620
BKW000630
BKW000640
BKW000650
BKW000660
BKW000670
BKW000680
BKW000690
BKW000700

C*** DETERMINE MONTE CARLO BACKWARD TRAVERSE ANGLES THETA (AT)
C*** AND PHI (BT) INSIDE DETECTOR CONE.
BT=TWOP*I*RAND( SEED)
AT=THV*RAND( SEED)
IF(AT.LE.1.E-30) AT=0.
IF(BT.LE.1.E-30) BT=0.
V=SIN(AT)
U23=COS(AT)
BKW000110
BKW000120
BKW000130
BKW000140
BKW000150
BKW000160
BKW000170
BKW000180
BKW000190
BKW000200
BKW000210
BKW000220
BKW000230
BKW000240
BKW000250
BKW000260
BKW000270
BKW000280
BKW000290
BKW000300
BKW000310
BKW000320
BKW000330
BKW000340
BKW000350
BKW000360
BKW000370
BKW000380
BKW000390
BKW000400
BKW000410
BKW000420
BKW000430
BKW000440
BKW000450
BKW000460
BKW000470
BKW000480
BKW000490
BKW000500
BKW000510
BKW000520
BKW000530
BKW000540
BKW000550
BKW000560
BKW000570
BKW000580
BKW000590
BKW000600
BKW000610
BKW000620
BKW000630
BKW000640
BKW000650
BKW000660
BKW000670
BKW000680
BKW000690
BKW000700

C*** ROTATE BACKWARD TRAVERSE VECTOR INTO STANDARD FRAME OF REFERENCE.
CALL ROTAT(AT,BT,R,SBC)
BKW000280
BKW000290
BKW000300
BKW000310
BKW000320
BKW000330
BKW000340
BKW000350
BKW000360
BKW000370
BKW000380
BKW000390
BKW000400
BKW000410
BKW000420
BKW000430
BKW000440
BKW000450
BKW000460
BKW000470
BKW000480
BKW000490
BKW000500
BKW000510
BKW000520
BKW000530
BKW000540
BKW000550
BKW000560
BKW000570
BKW000580
BKW000590
BKW000600
BKW000610
BKW000620
BKW000630
BKW000640
BKW000650
BKW000660
BKW000670
BKW000680
BKW000690
BKW000700

C*** DETERMINE BIASING DISTANCES FOR BACKWARD TRAVERSE.
CALL ELM(XD,SBC,EL)
BKW000310
BKW000320
BKW000330
BKW000340
BKW000350
BKW000360
BKW000370
BKW000380
BKW000390
BKW000400
BKW000410
BKW000420
BKW000430
BKW000440
BKW000450
BKW000460
BKW000470
BKW000480
BKW000490
BKW000500
BKW000510
BKW000520
BKW000530
BKW000540
BKW000550
BKW000560
BKW000570
BKW000580
BKW000590
BKW000600
BKW000610
BKW000620
BKW000630
BKW000640
BKW000650
BKW000660
BKW000670
BKW000680
BKW000690
BKW000700

C*** DETERMINE STATISTICAL WEIGHT REX OF BACKWARD TRAVERSE.
REX=1.-EXP(-GAMMA*(EL(2)-EL(1)))
REXRN=REX*RAND( SEED)
IF((1.-REXRN).LE.1.E-7) GO TO 15
BKW000340
BKW000350
BKW000360
BKW000370
BKW000380
BKW000390
BKW000400
BKW000410
BKW000420
BKW000430
BKW000440
BKW000450
BKW000460
BKW000470
BKW000480
BKW000490
BKW000500
BKW000510
BKW000520
BKW000530
BKW000540
BKW000550
BKW000560
BKW000570
BKW000580
BKW000590
BKW000600
BKW000610
BKW000620
BKW000630
BKW000640
BKW000650
BKW000660
BKW000670
BKW000680
BKW000690
BKW000700

C*** DETERMINE RANDOM DISTANCE FOR BIASED BACKWARD TRAVERSE.
ELBC=-ALOG(1.-REXRN)/GAMMA+EL(1)
CT=0.
DO 4 K=1,3
SBA(K)=XD(K)-XACK)+SBC(K)*ELBC
4 CT=CT+SBA(K)**2
IF(CT.LE.1.E-30) CT=0.
ELAB=SQRT(CT)
IF(ELAB.LT.ALIM)GO TO 15
BKW000410
BKW000420
BKW000430
BKW000440
BKW000450
BKW000460
BKW000470
BKW000480
BKW000490
BKW000500
BKW000510
BKW000520
BKW000530
BKW000540
BKW000550
BKW000560
BKW000570
BKW000580
BKW000590
BKW000600
BKW000610
BKW000620
BKW000630
BKW000640
BKW000650
BKW000660
BKW000670
BKW000680
BKW000690
BKW000700

C*** DETERMINE TIME BOX INDEX NT FOR THE COMPLETE BACKWARD TRAVERSE.
NT=1.+

```

```
ENC(NSCAP,NT)=ENC(NSCAP,NT)+DOM  
RETURN  
15 ENC(NSCAP)=ENC(NSCAP)-1.  
STH=0,  
RETURN  
END
```

```
BKW00710  
BKW00720  
BKW00730  
BKW00740  
BKW00750  
BKW00760
```

```

SUBROUTINE CONV(NP,NMS,NMAX,NSCA)           CON00010
C*****THIS SUBROUTINE CONVOLUTES A GIVEN SEQUENCE OF VALUES EN(NSCA,N),   CON00020
C N=1,NMS, WITH THE UNIT SQUARE FUNCTION EXTENDING FROM TIME=0 TO   CON00030
C*****TIME=(NP-1)*DELT.                   CON00040
DOUBLE PRECISION DBLE,XDBLE(100)             CON00050
COMMON/CONB/X(100),Y(100)                   CON00060
COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,   CON00070
+NTMAX,NSCAM,KMAX,LMAX,LMM1                CON00080
COMMON /BASBOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),    CON00090
+BEC(16),SINGWV,PF(65),LLMAX              CON00100
NMS1=NMS+1                                  CON00110
NMA=NMS+NP-1                                CON00120
XDBLE(1)=0.E-00                             CON00130
DO 100 N=2,NMS                            CON00140
100 XDBLE(N)=XDBLE(N-1)+DBLE(EN(NSCA,N-1))  CON00150
NP1=NP+1                                    CON00160
IF (NP.LT.NMS) GO TO 600                  CON00170
DO 200 N=1,NMS                            CON00180
200 Y(N)=XDBLE(N)                          CON00190
IF (NP.EQ.NMS) GO TO 400                  CON00200
DO 300 N=NMS1,NP                          CON00210
300 Y(N)=XDBLE(NMS)                      CON00220
400 CONTINUE                               CON00230
DO 500 N=NP1,NMA                         CON00240
500 Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)        CON00250
GO TO 1000                                CON00260
600 CONTINUE                               CON00270
DO 700 N=1,NP                           CON00280
700 Y(N)=XDBLE(N)                          CON00290
DO 800 N=NP1,NMS                         CON00300
800 Y(N)=XDBLE(N)-XDBLE(N-NP+1)          CON00310
DO 900 N=NMS1,NMA                         CON00320
900 Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)        CON00330
1000 CONTINUE                               CON00340
IF (NMA.EQ.NMAX) GO TO 1200               CON00350
NMA1=NMA+1                                CON00360
DO 1100 N=NMA1,NMAX                      CON00370
1100 Y(N)=0.0                                CON00380
DO 1150 N=1,NMAX                         CON00390
1150 EN(NSCA,N)=Y(N)                      CON00400
1200 RETURN                                CON00410
END                                     CON00420

```

```

SUBROUTINE ELM(X1,S1,EL)
C*** THIS SUBROUTINE DETERMINES THE SMALLEST (EL(1)) AND THE LARGEST
C*** (EL(2)) DISTANCES FROM A POINT X1() TO THE SURFACES OF AN
C*** ELLIPSOID ALONG A LINE OF SIGHT DEFINED BY UNIT VECTOR S1().
C
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
DIMENSION X1(3),S1(3),EL(2)
AT=0.
BT=0.
CT=-1.
DO 1 K=1,3
X=0.
Y=0.
DO 2 L=1,3
X=X+RE(K,L)*X1(L)
2 Y=Y+RE(K,L)*S1(L)
AT=AT+Y**2/ASQ(K)
BT=BT+X*Y/ASQ(K)
1 CT=CT+X**2/ASQ(K)
DISC=BT**2-AT*CT
IF(DISC.LT.-1.E-30)GO TO 10
DISC=SQRT(DISC)
EL(1)=- $(BT+DISC)/AT$ 
EL(2)= $(-BT+DISC)/AT$ 
IF(CT.LE.0.)EL(1)=0.
IF((EL(1).LT.0.).OR.(EL(2).LT.0.))GO TO 10
RETURN
10 EL(1)=0.
EL(2)=0.
RETURN
END

```

```

SUBROUTINE FIND(U1,PFU)
COMMON /M05/V1,C1,S1,SA(3),EN(10,100),EN(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASBOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
L=1
LL=LMM1
DO 1 K=1,KMAX
LL=LL/2
L=L+LL
AT=U1-U(L)
IF(AT.GT.1.E-7)L=L-LL
1 CONTINUE
PFU=PF(L)+(U1-U(L))*((PF(L+1)-PF(L))/(U(L+1)-U(L)))
RETURN
END

```

FIN00010
FIN00020
FIN00030
FIN00040
FIN00050
FIN00060
FIN00070
FIN00080
FIN00090
FIN00100
FIN00110
FIN00120
FIN00130
FIN00140
FIN00150
FIN00160



```

SUBROUTINE FWRD(SCA,DUM,ELAC) FWR00010
COMMON/CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK FWR00020
COMMON/FGEL/XA(3),D,NSCA FWR00030
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),AC(3) FWR00040
COMMON/RNDM/ SEED FWR00050
DIMENSION SCA(3) FWR00060
PSI=TWOP1*RAND < SEED> FWR00070
TH=PI2 FWR00080
CALL ROTAT(TH,PSI,R,SCA) FWR00090
ANR=RAND(SEED) FWR00100
IF(ANR.LE.1.E-30) ANR=0. FWR00110
RHO=AKM*SQRT(ANR) FWR00120
ELSQ=0. FWR00130
DO 1 K=1,3 FWR00140
SCA(K)=XD(K)-XA(K)+SCA(K)*RHO FWR00150
1 ELSQ=ELSQ+SCA(K)**2 FWR00160
IF(ELSQ.LE.1.E-30) ELSQ=1.E-30 FWR00170
ELAC=SQRT(ELSQ) FWR00180
DO 3 K=1,3 FWR00190
3 SCA(K)=SCA(K)/ELAC FWR00200
DUM=AKSQ/(4.*ELSQ) FWR00210
RETURN FWR00220
END FWR00230

```

```

SUBROUTINE GAS(JTYPE)                                GAS00010
C*** THIS SUBROUTINE DETERMINES A RANDOMLY-SELECTED SCATTERING   GAS00020
C*** DIRECTION USED IN PHOTON TRAVERSES WITHIN THE ELLIPSOIDAL   GAS00030
C*** AEROSOL CLOUD. JTYPE=1 SIGNIFIES THAT THE SCATTERING EVENT AT   GAS00040
C*** WHICH THIS ANGLE IS SELECTED IS WITHIN THE AEROSOL CLOUD.   GAS00050
C*** JTYPE=2 SIGNIFIES THAT THE SCATTERING EVENT IS ON THE GROUND   GAS00060
C*** PLANE.   GAS00070
C*** GAS00080
C*** GAS00090
COMMON/ALL/AT,BT,CT,BLIM                           GAS00100
COMMON/CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDECK      GAS00110
COMMON/M05/V1,C1,S1,SA(3),ENK(10,100),ENK(10),ELMIN,DELD,DTOT,   GAS00120
+NTMAX,NSCAM,KMAX,LMAX,LMMI                      GAS00130
COMMON/BASBOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),   GAS00140
+BEC(16),SINGWY,PF(65),LLMAX                   GAS00150
COMMON/RNDM/ SEED                               GAS00160
COMMON/FGEL/XAC(3),D,NSCA                         GAS00170
AT=RAND < SEED>                                GAS00180
IF(AT.LE.1.E-30) AT=0.                            GAS00190
IF(JTYPE.EQ.1) GO TO 1.                          GAS00200
SA(3)=1.-AT                                     GAS00210
ARG=1.-SA(3)**2                                  GAS00220
IF(ARG.LT.1.E-30) ARG=0.                         GAS00230
BT=TWOP1*RAND < SEED>                          GAS00240
IF(BT.LE.1.E-30) BT=0.                           GAS00250
V1=SQRT(ARG)                                    GAS00260
C1=COS(BT)                                     GAS00270
S1=SIN(BT)                                     GAS00280
SA(1)=V1*C1                                     GAS00290
SA(2)=V1*S1                                     GAS00300
RETURN                                         GAS00310
1 CALL USCA(AT,BT)                             GAS00320
CT=TWOP1*RAND < SEED>                          GAS00330
IF(CT.LE.1.E-30) CT=0.                           GAS00340
ARG=1.-BT**2                                    GAS00350
IF(ARG.LT.1.E-30) ARG=0.                         GAS00360
V=SQRT(ARG)                                    GAS00370
C=COS(CT)                                     GAS00380
S=SIN(CT)                                     GAS00390
SA(1)=BT*SA(1)+V*(C*C1*SA(3)-S*S1)           GAS00400
SA(2)=BT*SA(2)+V*(C*S1*SA(3)+S*C1)           GAS00410
SA(3)=BT*SA(3)-V*C*V1                          GAS00420
AT=SA(1)**2+SA(2)**2                           GAS00430
IF(AT.LT.1.E-10)GO TO 3.                        GAS00440
V1=SQRT(AT)                                    GAS00450
C1=SA(1)/V1                                    GAS00460
S1=SA(2)/V1                                    GAS00470
RETURN                                         GAS00480
3 C1=C
S1=S
V1=0.
SA(1)=0.
SA(2)=0;
BT=SA(3)
SA(3)=1.
IF(BT.LT.0.)SA(3)=-1.
RETURN
END

```

```

C      SUBROUTINE GMAX(IMAX,YMAX)
C*** THIS SUBROUTINE DETERMINES THE MAXIMUM VALUE YMAX OF AN INPUT
C*** ARRAY Y() OF DIMENSION IMAX.
C
COMMON/CONB/X(100),Y(100)
I=0
YMAX=Y(1)
1 I=I+1
IF(I.EQ.IMAX)RETURN
T=Y(I+1)-YMAX
IF(T.GT.0.)YMAX=Y(I+1)
GO TO 1
END

```

```

GMA00010
GMA00020
GMA00030
GMA00040
GMA00050
GMA00060
GMA00070
GMA00080
GMA00090
GMA00100
GMA00110
GMA00120
GMA00130
GMA00140

```

```

C SUBROUTINE MATRX(TH,PH,S,R)                                MAT00010
C*** THIS SUBROUTINE GENERATES UNIT VECTOR S() AND ROTATION MATRIX    MAT00020
C*** R() FOR A SET OF INPUT POLAR ANGLES (TH,PH).                      MAT00030
C*** THE ROTATION MATRIX R() ROTATES A VECTOR DEFINED RELATIVE TO THE    MAT00040
C*** <TH,PH> DIRECTION INTO THE STANDARD SYSTEM OF COORDINATES.          MAT00050
C*** THE UNIT VECTOR S() POINTS IN THE <TH,PH> DIRECTION IN THE          MAT00060
C*** STANDARD SYSTEM OF COORDINATES.                                     MAT00070
C                                         MAT00080
C                                         MAT00090
C COMMON /CONST/PI,PI2,PIRAD,TWOPi,TORRMB,CDECK
C DIMENSION S(3),R(3,3)
C AT=PIRAD*TH
C BT=PIRAD*PH
C IF(AT.LE.1.E-30) AT=0.
C IF(BT.LE.1.E-30) BT=0.
C V1=SIN(AT)
C C1=COS(AT)
C S1=SIN(BT)
C S(1)=V1*C1
C S(2)=V1*S1
C S(3)=COS(AT)
C R(1,1)=C1*S(3)
C R(1,2)=S1*S(3)
C R(1,3)=-V1
C R(2,1)=-S1
C R(2,2)=C1
C R(2,3)=0.
C DO 1 K=1,3
C   R(3,K)=S(K)
C RETURN
C END
                                         MAT00100
                                         MAT00110
                                         MAT00120
                                         MAT00130
                                         MAT00140
                                         MAT00150
                                         MAT00160
                                         MAT00170
                                         MAT00180
                                         MAT00190
                                         MAT00200
                                         MAT00210
                                         MAT00220
                                         MAT00230
                                         MAT00240
                                         MAT00250
                                         MAT00260
                                         MAT00270
                                         MAT00280
                                         MAT00290
                                         MAT00300
                                         MAT00310

```

```

SUBROUTINE ROTAT(TH,PH,R,S)           ROT00010
DIMENSION R(3,3),X(3),S(3)           ROT00020
IF(TH.LE.1.E-30) TH=0.                ROT00030
IF(PH.LE.1.E-30) PH=0.                ROT00040
V=SIN(TH)                           ROT00050
X(1)=V*COS(PH)                      ROT00060
X(2)=V*SIN(PH)                      ROT00070
X(3)=COS(TH)                        ROT00080
DO 1 J=1,3                           ROT00090
S(J)=0.                             ROT00100
DO 1 K=1,3                           ROT00110
1 S(J)=S(J)+R(K,J)*X(K)            ROT00120
RETURN                               ROT00130
END                                  ROT00140

```

```

SUBROUTINE SMOOZ(NSCA,NO)           SMO00010
C*****THIS SUBROUTINE DETERMINES 'NO', THAT VALUE OF I BEYOND WHICH ALL   SMO00020
C*****Y(I) IN A SEQUENCE ARE ZERO.                                         SMO00030
COMMON /MOS/Y1,L1,S1,SA(3),EN(10,100),EN(100),ELMIN,DELD,DTOT,      SMO00040
+NTMAX,NSCAM,KMAX,LMAX,LMM1                                              SMO00050
COMMON /BASBOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),          SMO00060
+BE(16),SINGWY,PF(65),LLMAX                                             SMO00070
I=NTMAX+1
100 I=I-1
    IF(I.EQ.0)GO TO 200
    YI=EN(NSCA,I)
    IF (YI.GT.0.0) GO TO 200
    GO TO 100
200 NO=I
    RETURN
    END

```

```

C      SUBROUTINE START(SS)                      STA00010
C*** THIS SUBROUTINE INITIALIZES PHOTON LAUNCH DIRECTION AND CALCULATES STA00030
C*** DIRECT BEAM (ZEROTH ORDER) CONTRIBUTIONS TO RECEIVED POWER.          ST. 00040
C
C      COMMON/ALL/AT,BT,CT,BLIM                      STA00050
C      COMMON/CONB/XC(100),YC(100)                   STA00060
C      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)    STA00070
C      COMMON/HIT/UDS,THSP,RS(3,3),XSC(3),DSA,XV(3)           STA00080
C      COMMON/FGEL/XAC(3),D,NSCA                     STA00090
C      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THY,TAU,EL(2), STA00100
*ALB(2),ZG,DMAX                         STA00110
C      COMMON/RNDM/ SEED                          STA00120
C      COMMON /M05/V1,C1,S1,SA(3),EN(10,100),EN(10),ELMIN,DELD,DTOT, STA00130
+NTMAX,NSCAM,KMAX,LMAX,LMM1             STA00140
C      COMMON /BASBOT/UK(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16), STA00150
+BE(16),SINGWV,PFC(65),LLMAX          STA00160
C      COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK        STA00170
C      DIMENSION SS(3)                           STA00180
C
C      GENERATE RANDOM THETA (AT) AND PHI (BT) PHOTON LAUNCH ANGLES STA00190
C*** CONFINED WITHIN SOURCE CONE.          STA00200
C
C      ANR=RAND(SEED)                          STA00210
C      IF(ANR.LE.1.E-30) ANR=0.                 STA00220
C      AT=THSP*SQRT(ANR)                      STA00230
C      BT=TWOPI*RAND ( SEED)                  STA00240
C
C*** ROTATE LAUNCH VECTOR SA( ) INTO STANDARD FRAME OF REFERENCE. STA00250
C
C      CALL ROTAT(AT,BT,RS,SA)                STA00260
C
C*** DETERMINE HORIZONTAL COMPONENT CT OF LAUNCH VECTOR AND BRANCH STA00270
C*** TO VERTICAL TREATMENT (WHICH INCLUDES GROUND PLANE) IF THIS STA00280
C*** COMPONENT IS VERY SMALL.            STA00290
C
C      CT=SA(1)**2+SA(2)**2                 STA00300
IF(CT.LE.1.E-24)GO TO 1               STA00310
C      V1=SQRT(CT)                          STA00320
C1=SA(1)/V1                           STA00330
S1=SA(2)/V1                           STA00340
GO TO 2                                STA00350
C
C*** DEFINE VERTICAL UNIT VECTOR          STA00360
C
1 V1=0,                                STA00370
CT=SA(3)                               STA00380
SA(3)=1                                STA00390
IF((CT.LT.0.) .AND.(ALB(2).GT.0.)) SA(3)=-1. STA00400
C1=1,                                   STA00410
S1=0,                                   STA00420
SA(1)=0,                                 STA00430
SA(2)=0.                                STA00440
C
C*** INITIALIZE TOTAL TRAVERSE DISTANCE DTOT AND STATISTICAL STA00450
C*** STRENGTH STH.                      STA00460
C
2 DTOT=0,                                STA00470
STH=1,                                   STA00480
BT=0,                                    STA00490
IF(AT.LE.1.E-30) GO TO 11              STA00500
DAA=DSA/COS(AT)                        STA00510
DAD=0,                                   STA00520
DO 3 K=1,3                            STA00530
XACK)=XV(K)+DAA*SA(K)                 STA00540
DAD=DAD+Y(K)**2                        STA00550
3 BT=BT+Y(K)*SS(K)                     STA00560
IF(DAD.LE.1.E-30) DAD=0.                 STA00570
DAD=SQRT(DAD)                          STA00580
C

```

```

C*** DETERMINE DIRECT BEAM ATTENUATION DISTANCE AND BIASING DISTANCES STA00710
C*** FOR INITIAL PHOTON TRAVERSE. STA00720
C STA00730
C STA00740
C STA00750
C STA00760
C STA00770
C STA00780
C STA00790
C STA00800
C STA00810
C STA00820
C STA00830
C STA00840
C STA00850
C STA00860
C STA00870
C STA00880
C STA00890
C STA00900
C STA00910
C STA00920
C STA00930
C STA00940
C STA00950
C STA00960
C STA00970
C STA00980
C STA00990
C STA01000
C STA01010
C STA01020
C STA01030
C STA01040
C STA01050
C
      CALL ELM(XA,SA,EL)
      IF(BT.LE.ALIM) GO TO 4
      IF(DAD.LE.(1.001*DSA)) GO TO 4
      BT=0,
      CT=0,
      DO 5 K=1,3
      CT=CT+SA(K)*SD(K)
      5 BT=BT+Y(K)*SD(K)
      CTSQ=CT**2
      IF(CTSQ.LE.1.E-30) GO TO 4
      ELSD=ABS(BT/CT)-DAA
      CT=0,
      DO 6 K=1,3
      6 CT=CT+(XA(K)+SA(K)*ELSD-XD(K))**2
      IF(CT.GT.ARSQ) GO TO 4
      IF(EL(2).LE.EL(1)) STH=0,
      IF(ELD(1).LE.0.) EL(2)=ELSD
      IF(DAD.LT.ALIM) GO TO 4
      UDS=-BT/DAD
      IF(UDS.LT.UV)RETURN
      NT=1.+ $(ELSD-ELMIN)/DELD$ 
      IF(NT.LT.0) GO TO 11
      IF(NT.EQ.0) NT=1
      EN(1,NT)=EN(1,NT)+EXP(-GAMMA*(EL(2)-EL(1)))
      RETURN
      4 CONTINUE
      IF((EL(2)-EL(1)).GT.1.E-20) RETURN
      IF((SA(3).GE.0.).OR.(ALB(2).LE.1.E-20)) STH=0.
      RETURN
      11 STH=0.
      RETURN
      END

```

```

SUBROUTINE TRAVRS( JTYPE, I2FLG, ICOND)
COMMON/RNDM/ SEED
COMMON/ALL/AT,BT,CT,BLIM
COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
*ALB(2),ZG,DMAX
COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASBOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
COMMON/FGEI/XAC(3),D,NSCA
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
COMMON/CONST/P1,P12,PIRAD,TWOP1,TORRMB,CDEGK
COMMON/CONB/X(100),Y(100)
DIMENSION SDA(3)
DGAM=EL(2)-EL(1)
REXIT=.99999
RX=.99999
RXX=.99999
IF((DGAM.LE.0.),AND.(ALB(2).GT.0.))GO TO 9
IF(DGAM.LE.1.E-7) GO TO 70
IF(I2FLG.EQ.1) GO TO 5
IF((SA(3).GE.0.),OR.(ALB(2).LE.0.))REXIT=1.-EXP(-GAMMA*DGM)
AT=EL(1)-(ALB(1)-REXIT*RAND('SEED'))/GAMMA
GO TO 4
5 CONTINUE
IF(ELD(1).GT.ALIM)ALIM=ELD(1)
RXX=(1./GAMMA)*((1./ALIM)-(1./DGAM))
DENOM=<(1./ALIM)-GAMMA*RXX*RAND(SEED)>
IF(DENOM.LE.1.E-7) GO TO 70
AT=1./DENOM
RX=EXP(-GAMMA*AT)*(AT*GAMMA)**2
BT=XAC(3)+SA(3)*AT
IF((BT.GT.ZG).AND.(AT.LE.EL(2)))GO TO 2
9 JTYPE=2
AT=(ZG-XAC(3))/SA(3)
GO TO 1
2 JTYPE=1
1 STH=ALB(JTYPE)*REXIT*STH*RXX
IF(STH.LE.0.)RETURN
DO 3 K=1,3
3 XACK=XACK+AT*SACK
DTOT=DTOT+AT
IF(NSCA.NE.2) GO TO 50
AT=0,
DO 8 K=1,3
SDACK=SDACK-XACK
8 AT=AT+SDACK*SDACK
IF(AT.LT.BLIM)GO TO 15
AT=SQRT(AT)
BT=0,
DO 32 K=1,3
SDACK=SDACK/AT
32 BT=BT-SDACK*SDACK
IF(BT.LT.UV) GO TO 50
CALL FWRD(SDA,DOM,AT)
NT=1.+ $(DTOT+AT-ELMIN)/DELD$ 
IF(NT.LT.0) RETURN
IF(NT.EQ.0) NT=1
IF((NT.EQ.1).AND.(I2FLG.EQ.0).AND.(ICOND.EQ.1))GO TO 50
IF(NT.GT.NTMAX)RETURN
CT=0,
BT=0,
DO 7 K=1,3
BT=BT-SDACK*SDACK
7 CT=CT+SDACK*SACK
PFI=1,
IF(JTYPE.EQ.1)CALL FIND(CT,PFI)
CALL ELM(XA,SDA,EL)
AT=AT-EL(1)
IF(ELD(1).GT.0.)AT=EL(2)-EL(1)

```

DOM=DOM*PFI*STH*EXP(-GAMMA*AT)*BT*RX	TRA00710
IF(I2FLG.NE.1) GO TO 13	TRA00720
IF(NT.EQ.1) DOM=DOM*10.	TRA00730
IF(NT.GT.1) DOM=0.	TRA00740
GO TO 14	TRA00750
13 CONTINUE	TRA00760
IF(ICOND.EQ.1) DOM=DOM/0.9	TRA00770
14 EN(NSCA,NT)=EN(NSCA,NT)+DOM	TRA00780
GO TO 50	TRA00790
15 ENC(2)=ENC(2)-1.	TRA00800
STH=0.	TRA00810
RETURN	TRA00820
50 CONTINUE	TRA00830
IF(I2FLG.EQ.1) GO TO 60	TRA00840
CALL BKWD(JTYPE)	TRA00850
60 RETURN	TRA00860
70 ENC(NSCA)=ENC(NSCA)-1.	TRA00870
STH=0.	TRA00880
RETURN	TRA00890
END	TRA00900

```

SUBROUTINE USCA(SC,US)
COMMON /MQ5/V1,C1,S1,SAC(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASBOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
L=1
LL=LMM1
DO 1 K=1,KMAX
LL=LL/2
L=L+LL
IF(SUM(L).GT.SC) L=L-LL
1 CONTINUE
US=U(L)+(SC-SUM(L))*(U(L+1)-U(L))/(SUM(L+1)-SUM(L))
RETURN
END

```

USC00010
 USC00020
 USC00030
 USC00040
 USC00050
 USC00060
 USC00070
 USC00080
 USC00090
 USC00100
 USC00110
 USC00120
 USC00130
 USC00140
 USC00150

```

SUBROUTINE SMOKE(WAVE1,ICLMAT,TRANS,IERR) SMK00010
COMMON /IOUNIT/IOUTH,IOUTL,IPHFUN,LOUNT,NDIRTU,NCLIMT,KSTOR,NPLOTUSMK00021
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT, SMK00031
1 FOGPRB,WNDVEL,WNDDIR,IPASCT SMK00040
COMMON /GEOMET/ PTS(15),IGEOSW SMK00050
COMMON /CONST/PI,PI2,PIRAD,TWOPi,TORRMB,CDEGK SMK00060
DIMENSION IR(26),ITTL(11,5),TSUB(11),QDIV(11) SMK00070
EXTERNAL JPASCT SMK00080

```

C***** NEW PROGRAM OPTIONS ADDED AS REVISIONS 03 & 04

THIS REVISION TO THE MODEL ALLOWS THE OPTION OF PRODUCING A ONE DIMENSIONAL "SNAPSHOT" OF A SMOKE SCREEN DUE TO ANY NUMBER OF MUNITIONS AT SOME SPECIFIED TIME; OUTPUT FOR THIS OPTION IS CROSSWIND TRANSMISSION AS A FUNCTION OF DOWNWIND DISTANCE AT A SINGLE GIVEN TIME RATHER THAN TRANSMISSION AS A FUNCTION OF TIME AT A SINGLE GIVEN LINE OF SIGHT.

NEW INPUTS ARE ENTERED THROUGH THE "NAME" CARD AS:

NAME
 STIME = SINGLE GIVEN TIME AT WHICH SCREEN IS TO BE SAMPLED
 FRONT = LENGTH OF SCREEN TO BE SAMPLED (ALONGWIND)
 DELX = INCREMENTS BETWEEN CONTIGUOUS LINES OF SIGHT
 MCUPRT = OPTION TO SUPPRESS INTERMEDIATE OUTPUT (1 = SUPPRESS)

ALSO THE TIME OF DETONATION FOR EACH MUNITION IS REQUIRED (IN SECONDS) AS THE FOURTH ENTRY ON THE "MUNC" CARD

C***** THE CAVEATS FOR THE NEW OPTION ARE:

- 1) THE LINES OF SIGHT MUST BE CROSSWIND
- 2) STARTING POINT FOR SAMPLING IS THE OBSERVER-TARGET COORDINATES ENTERED ON THE "OBSC" AND "TARC" CARDS
- 3) THE PRNT OPTION ON THE "OUTP" CARD MUST BE ZERO
- 4) A PLOT FILE OPTION HAS BEEN ADDED AS THE FOURTH ENTRY ON THE "OUTP" CARD (NPLT=1 WILL CREATE A PLOT FILE OF THE FINAL RESULTS ON UNIT NPLOTU - SEE EOMAIN)

NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC) TO ALTER PERIOD, CHANGE MAXS AND DIMENSIONS OF SMAS, PVOL AND CLTOT.

```

COMMON /M05/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),
*EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)
DATA MAXS/960/
DATA QDIV/.0.,.3,.50,.60,.65,.70,.80,1.0,1.,1.,1./

```

NOTE: FOR SHORTER OR LONGER BURNS AND OBSCURATIONS, USERS MAY SUBSTITUTE FOR THE COMMON BLOCK AND MAXS PARAMETER AND ALTER DIMENSIONS FOR SMAS, PVOL AND CLTOT.

SMOKE MODEL

FORMAL INPUTS:

```

WAVE1 = WAVELENGTH IN MICROMETERS (OR 94. GHZ) FROM ALLOWED BANDS (SEE EXTC RECORD BELOW). SMK00260
ICLMAT = CLIMATOLOGY FLAG. IF 1 THEN /CLYMAT/ VALUES OVER-RIDE METR RECORD VALUES. SMK00270
IGEOSW = GEOMETRY FLAG FROM COMMON GEOMET. IF SET TO 1 IN EOMAIN, THEN OBSERVER AND TARGET COORDINATES ARE PASSED TO SMOKE, AND ANGLE XNORTH SET TO 90. DEGREES. SMK00290

```

FORMAL OUTPUTS:

```

TRANS = TRANSMISSION AT TIME ETO (DEFAULT) OR USER SPECIFIED TIME (SEE OUTP RECORD BELOW) FOR WAVELENGTH WAVE1 ALONG THE OBSERVER-TARGET LINE-OF-SIGHT. SMK00350

```

IERR = ERROR FLAG. 1 IF ERROR IN SMOKE. SMK00390

USER RECORDS INPUT:
 EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,
 FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.

```

SMK00010
SMK00021
SMK00031
SMK00040
SMK00050
SMK00060
SMK00070
SMK00080

```

```

SMK00090
SMK00100
SMK00110
SMK00120
SMK00130
SMK00140
SMK00150
SMK00160
SMK00170
SMK00180
SMK00190
SMK00200
SMK00210
SMK00220
SMK00230
SMK00240
SMK00250
SMK00260
SMK00270
SMK00280
SMK00290
SMK00300
SMK00310
SMK00320
SMK00330
SMK00340
SMK00350
SMK00360
SMK00370
SMK00380
SMK00390
SMK00400
SMK00410
SMK00420

```

PER FIELD BEGINNING IN COL 11. THE CARDS ARE NOT ORDER DEPENDENT ALTHOUGH VALUES ON EACH RECORD MUST FOLLOW THE ORDER SHOWN IN COMMENTS BELOW. FORMAT (2A2,6X,7F10.3)

THE FOLLOWING ARE REQUIRED RECORDS FOR AT LEAST ONE INPUT SET.

IDENT.	VARS.	DESCRIPTION	
MUNC	XM,YM,ZM,TM (M,M,M,SEC)	MUNITION COORDINATES AND EVENT TIME	SMK00430 SMK00440 SMK00450 SMK00460 SMK00470 SMK00480 SMK00490 SMK00500 SMK00510 SMK00520
BART	STO ETO DTO ANGLX	COMPUTATION TIMES AND X-AXIS DEFINITION: OUTPUT STARTING TIME (SEC. SINCE IGNITION) ENDING TIME FOR CALCULATION (SEC) TIME INCREMENT FOR OUTPUT TABLES (SEC) ANGLE OF POSITIVE X-AXIS WRT NORTH <DEG. CLOCKWISE WRT NORTH> ASSUMED 90 DEG. IF IGEOSW IS 1 FROM EOMAIN	SMK00530 SMK00540 SMK00550 SMK00560 SMK00570 SMK00580 SMK00590 SMK00600 SMK00610 SMK00620
OPTIONAL DEPENDING ON PARAMETERS CHOSEN:			SMK00630 SMK00640 SMK00650 SMK00660 SMK00670 SMK00680 SMK00690
OBSC	XO,YO,ZO	OBSERVER COORDINATES (IGNORED IF IGEOSW IS 1 FROM EOMAIN). (M,M,M)	SMK00700 SMK00710 SMK00720
TARC	XT,YT,ZT	TARGET COORDINATES (IGNORED IF IGEOSW IS 1 FROM EOMAIN) (M,M,M)	SMK00730 SMK00740 SMK00750 SMK00760 SMK00770 SMK00780
OUTP		OPTIONAL RECORD TO SELECT AMOUNT OF PRINT OUTPUT AND TO SELECT CRITERIA FOR RETURNED TRANSMISSION.	SMK00790 SMK00800 SMK00810 SMK00820 SMK00830 SMK00840 SMK00850 SMK00860 SMK00870 SMK00880
PRNT		IF 0., ALL FULL OUTPUT LISTINGS CREATED. IF 1., IS ENTERED AT ANY POINT, THEN ALL FURTHER OUTPUT IS SUPPRESSED, EXCEPT FOR THE FINAL ACCUMULATED EFFECTS LISTING OF TOTAL CL AND TOTAL TRANSMITTANCE FOR COMBINED MULTIPLE INPUT SETS. (DEFAULT IS PRNT = 0.)	SMK00890 SMK00900 SMK00910 SMK00920
CRITER		SELECTS CHOICE OF TRANSMISSION RETURNED FROM SMOKE. 0. = RETURN TOTAL TRANSMITTANCE COMPUTED AT LAST TIME ETO. (DEFAULT CASE) 1. = RETURN THE MINIMUM VALUE OF TOTAL TRANSMITTANCE COMPUTED FOR WAVELENGTH WAVE1. 2. = RETURN VALUE OF TOTAL TRANSMITTANCE FOR WAVELENGTH WAVE1 COMPUTED AT USER- SPECIFIED TIME TIMTRN BELOW.	SMK00930 SMK00940 SMK00950 SMK00960 SMK00970 SMK00980
TIMTRN		REQUIRED ONLY IF CRITER IS 2. TIME FOR WHICH TRANSMITTANCE RETURNED IS COMPUTED. SHOULD BE CLOSE OR EQUAL TO A TABLE TIME (AS DETERMINED BY THE BART RECORD) FOR ACCURACY.	SMK00990 SMK01000 SMK01010 SMK01020 SMK01030 SMK01040
IPLT		PLOT CODE ADDED IN ORDER TO PLOT OUTPUT ON UNIT NPLOTU IF IPLT = 1.	SMK01050 SMK01060 SMK01070 SMK01080 SMK01090
MUNT		REQUIRED IF BURN RECORD IS NOT USED. OTHERWISE, OPTIONAL. ANY NON-ZERO VALUES INPUT WILL OVER- RIDE PREVIOUS SOURCE DEFINITIONS (INCLUDING THOSE FROM THE BURN RECORD.)	SMK01090 SMK01100 SMK01110 SMK01120 SMK01130 SMK01140
XN		NUMBER OF MUNITIONS IGNITED AT THE SAME LOCATION AND AT THE SAME TIME (DIMENSIONLESS)	SMK01150 SMK01160 SMK01170 SMK01180 SMK01190 SMK01200
FW		FILL WEIGHT OF ONE MUNITION (LBS.) FOR WP, PWP, HC OR RP (BUT RATE OF BURN IN GAL./HR. FOR FOG OIL. NOTE 1 GAL/HR=	SMK01210 SMK01220 SMK01230 SMK01240 SMK01250 SMK01260
TBURN TYPE		BURN DURATION FOR THIS MUNITION (SEC) TYPE OF SMOKE (DIMENSIONLESS)	SMK01270 SMK01280 SMK01290 SMK01300 SMK01310 SMK01320
EFF		1.=WP, 2.=PWP OR WP WICK/WEDGE, 3.=HC, 4.=FOG OIL, 5.=RP MUNITION BURN EFFICIENCY. (PERCENT)	SMK01330 SMK01340 SMK01350 SMK01360 SMK01370 SMK01380

YF	IF INPUT AS 0., THEN DEFAULT IS USED. YIELD FACTOR (DIMENSIONLESS), IF 0., DEFAULT TO RELATIVE HUMIDITY DEPENDENT STRAIGHT LINE FIT FOR WP, PWP, RP FROM JOHNSON AND FORNEY. SIMILARLY FOR HC. FOG OIL IS SET TO 1.	SMK01100 SMK01110 SMK01120 SMK01130 SMK01140 SMK01150 SMK01160 SMK01170 SMK01180 SMK01190 SMX01200 SMK01210 SMK01220 SMK01230 SMK01240 SMK01250 SMK01260 SMK01270 SMK01280 SMK01290 SMK01300 SMK01310 SMK01320 SMK01330 SMK01340 SMK01350 SMK01360 SMK01370 SMK01380 SMK01390 SMK01400 SMK01410 SMK01420 SMK01430 SMK01440 SMK01450 SMK01460 SMK01470 SMK01480 SMK01490 SMK01500 SMK01510 SMK01520 SMK01530 SMK01540 SMK01550 SMK01560 SMK01570 SMK01580 SMK01590 SMK01600 SMK01610 SMK01620 SMK01630 SMK01640 SMK01650 SMK01660 SMK01670 SMK01680 SMK01690 SMK01700 SMK01710 SMK01720 SMK01730 SMK01740 SMK01750 SMK01760 SMK01770 SMK01780 SMK01790
METR	REQUIRED IF ICLMAT IS ZERO. OTHERWISE, NEEDED ONLY FOR E OR F PASQUILL CATEGORY TO PROVIDE TGRAD WHICH IS NOT AVAILABLE IN CLYMAT.	
RELHUM UW WNDIR	RELATIVE HUMIDITY (PERCENT) WIND VELOCITY (M/SEC) WIND DIRECTION (USUAL MET CONVENTION, ANGLE IN DEG. CLOCKWISE FROM NORTH OF DIRECTION FROM WHICH WIND ORIGINATES.)	SMK01200 SMK01210 SMK01220 SMK01230 SMK01240 SMK01250 SMK01260 SMK01270 SMK01280 SMK01290 SMK01300 SMK01310 SMK01320 SMK01330 SMK01340 SMK01350 SMK01360 SMK01370 SMK01380 SMK01390 SMK01400 SMK01410 SMK01420 SMK01430 SMK01440 SMK01450 SMK01460 SMK01470 SMK01480 SMK01490 SMK01500 SMK01510 SMK01520 SMK01530 SMK01540 SMK01550 SMK01560 SMK01570 SMK01580 SMK01590 SMK01600 SMK01610 SMK01620 SMK01630 SMK01640 SMK01650 SMK01660 SMK01670 SMK01680 SMK01690 SMK01700 SMK01710 SMK01720 SMK01730 SMK01740 SMK01750 SMK01760 SMK01770 SMK01780 SMK01790
PCAT	PASQUILL CATEGORY (DIMENSIONLESS)	SMK01200 SMK01210 SMK01220 SMK01230 SMK01240 SMK01250 SMK01260 SMK01270 SMK01280 SMK01290 SMK01300 SMK01310 SMK01320 SMK01330 SMK01340 SMK01350 SMK01360 SMK01370 SMK01380 SMK01390 SMK01400 SMK01410 SMK01420 SMK01430 SMK01440 SMK01450 SMK01460 SMK01470 SMK01480 SMK01490 SMK01500 SMK01510 SMK01520 SMK01530 SMK01540 SMK01550 SMK01560 SMK01570 SMK01580 SMK01590 SMK01600 SMK01610 SMK01620 SMK01630 SMK01640 SMK01650 SMK01660 SMK01670 SMK01680 SMK01690 SMK01700 SMK01710 SMK01720 SMK01730 SMK01740 SMK01750 SMK01760 SMK01770 SMK01780 SMK01790
AIRT TGRAD	SURFACE AIR TEMPERATURE (DEG C) VERT TEMP GRADIENT (C DEG/M). EXAMPLE: TGRAD=(AIRT(10 M)-AIRT(.5 M))/9.5 M (USED ONLY FOR PASQUILL CATEGORIES E, F)	SMK01200 SMK01210 SMK01220 SMK01230 SMK01240 SMK01250 SMK01260 SMK01270 SMK01280 SMK01290 SMK01300 SMK01310 SMK01320 SMK01330 SMK01340 SMK01350 SMK01360 SMK01370 SMK01380 SMK01390 SMK01400 SMK01410 SMK01420 SMK01430 SMK01440 SMK01450 SMK01460 SMK01470 SMK01480 SMK01490 SMK01500 SMK01510 SMK01520 SMK01530 SMK01540 SMK01550 SMK01560 SMK01570 SMK01580 SMK01590 SMK01600 SMK01610 SMK01620 SMK01630 SMK01640 SMK01650 SMK01660 SMK01670 SMK01680 SMK01690 SMK01700 SMK01710 SMK01720 SMK01730 SMK01740 SMK01750 SMK01760 SMK01770 SMK01780 SMK01790
EXTC	OPTIONAL USER OVERRIDE FOR EXTINGUISH COEFFICIENTS. IF RECORD NOT USED, OR FOR ANY VALUES READ IN AS 0., THE EXTINGUISH COEFF. DEFAULTS TO ALPHA ARRAY VALUE IN STRANS.	SMK01200 SMK01210 SMK01220 SMK01230 SMK01240 SMK01250 SMK01260 SMK01270 SMK01280 SMK01290 SMK01300 SMK01310 SMK01320 SMK01330 SMK01340 SMK01350 SMK01360 SMK01370 SMK01380 SMK01390 SMK01400 SMK01410 SMK01420 SMK01430 SMK01440 SMK01450 SMK01460 SMK01470 SMK01480 SMK01490 SMK01500 SMK01510 SMK01520 SMK01530 SMK01540 SMK01550 SMK01560 SMK01570 SMK01580 SMK01590 SMK01600 SMK01610 SMK01620 SMK01630 SMK01640 SMK01650 SMK01660 SMK01670 SMK01680 SMK01690 SMK01700 SMK01710 SMK01720 SMK01730 SMK01740 SMK01750 SMK01760 SMK01770 SMK01780 SMK01790
BURN	OPTIONAL - SELECTS BUILT-IN MUNITION CHARACTERISTICS FROM THE BRATE ROUTINE FOR ONE (XN=1) MUNITION. VALUES ARE FOR FILL WEIGHT (FW), BURN DURATION (TBURN), SMOKE TYPE (CTYPE), EFFICIENCY (EFF). YIELD FACTOR IS SET TO ZERO SO THAT RH MODEL DEPENDENT VALUES ARE USED. ANY VALUES READ IN AS NON-ZERO ON A MUNT RECORD (WHICH IS OPTIONAL IF A BURN CARD IS USED) WILL OVERRIDE THE DEFAULTS STORED IN BRATE.	SMK01200 SMK01210 SMK01220 SMK01230 SMK01240 SMK01250 SMK01260 SMK01270 SMK01280 SMK01290 SMK01300 SMK01310 SMK01320 SMK01330 SMK01340 SMK01350 SMK01360 SMK01370 SMK01380 SMK01390 SMK01400 SMK01410 SMK01420 SMK01430 SMK01440 SMK01450 SMK01460 SMK01470 SMK01480 SMK01490 SMK01500 SMK01510 SMK01520 SMK01530 SMK01540 SMK01550 SMK01560 SMK01570 SMK01580 SMK01590 SMK01600 SMK01610 SMK01620 SMK01630 SMK01640 SMK01650 SMK01660 SMK01670 SMK01680 SMK01690 SMK01700 SMK01710 SMK01720 SMK01730 SMK01740 SMK01750 SMK01760 SMK01770 SMK01780 SMK01790
TYPM	MUNITION TYPE: 0. = USER DEFINED MUHITION SOURCE CHAR. 1. = 155MM HC, M1 CANISTER. 2. = 155MM HC, M2 CANISTER. 3. = 105MM HC CANISTER. 4. = 155MM HC M116B1 PROJ. 5. = 105MM HC M84A1 PROJ. 6. = SMOKE POT HC M5 7. = 60MM WP M302 CARTRIDGE 8. = 81MM WP M375A2 9. = 4.2 IN WP M326A1 10. = 155MM WP M110E2 11. = 105MM WP M60A2 12. = 4.2 IN PWP M328A1 13. = 5. IN PWP ZUNI MK4 14. = 2.75 IN WP WEDGE SUB-MUNITION. 15. = 3. IN WP WICK SUB-MUNITION 16. = 6. IN WP WICK SUB-MUNITION 17. = 155MM WP WEDGE XM825 (.92 SUB-MUN.) 18. = 81MM RP WEDGE NAVY SUB-MUNITION	SMK01200 SMK01210 SMK01220 SMK01230 SMK01240 SMK01250 SMK01260 SMK01270 SMK01280 SMK01290 SMK01300 SMK01310 SMK01320 SMK01330 SMK01340 SMK01350 SMK01360 SMK01370 SMK01380 SMK01390 SMK01400 SMK01410 SMK01420 SMK01430 SMK01440 SMK01450 SMK01460 SMK01470 SMK01480 SMK01490 SMK01500 SMK01510 SMK01520 SMK01530 SMK01540 SMK01550 SMK01560 SMK01570 SMK01580 SMK01590 SMK01600 SMK01610 SMK01620 SMK01630 SMK01640 SMK01650 SMK01660 SMK01670 SMK01680 SMK01690 SMK01700 SMK01710 SMK01720 SMK01730 SMK01740 SMK01750 SMK01760 SMK01770 SMK01780 SMK01790

	19. = 81MM RP WEDGE GERMAN SUB-MUNITION	SMK01800
	20. = 155MM RP WEDGE XM803 (228 SUB-MUN.)	SMK01810
	21. = FOG OIL GENERATOR (EXAMPLE)	SMK01820
BRAT1	BURN-RATE COEFF. (CONST. TERM)	SMK01830
BRAT2	BURN-RATE COEFF. (LINEAR TERM)	SMK01840
BRAT3	BURN-RATE COEFF. (QUADRATIC TERM)	SMK01850
BRAT4	BURN-RATE COEFF. (CUBIC TERM) RATE COEFF. ARE BASED ON THE SCALED POLYNOMIAL DESCRIBING THE FRACTIONAL MASS BURNED AT SCALED TIMES FROM 0. (IGNITION) TO 1. (BURN-OUT). THE INTEGRAL OF THE POLYNOMIAL FROM 0. TO 1. MUST BE 1., IE. THE SCALED TOTAL MASS BURNED BY TIME 1. IS 100 PERCENT. THE PROGRAM WILL CHECK FOR THIS CONDITION AND RE-SCALE COEFF. IF NECESSARY.	SMK01860 SMK01870 SMK01880 SMK01890 SMK01900 SMK01910 SMK01920 SMK01930 SMK01940 SMK01950
BRAT5	SPECIAL BURN RATE COEFFICIENT FOR ESTIMATING THE RAPID BURST FOLLOWED BY SLOW BURN OF SOME MUNITIONS. THE FUNCTIONAL FORM OF THE TERM MULTIPLIED BY BRAT5 IS	SMK01960 SMK01970 SMK01980 SMK01990
.07818	* < 358800. / < 1. + 358800. T/TBURN >> THOSE MUNITIONS FOR WHICH THE RATE IS NOT MODELED WILL RECEIVE BRAT1=1, BRAT2,3,4,5 = 0 (IE. CONSTANT BURN-RATE). THE USER MAY OVERIDE THE DEFAULT RATES BY ENTERING VALUES ON THE BURN CARD.	SMK02000 SMK02010 SMK02020 SMK02030 SMK02040 SMK02050
NAME	SCREEN PARAMETERS	
STIME	TIME AT WHICH SCREEN IS EXAMINED	
FRONT	SCREEN FRONTAGE TO BE EXAMINED	
DELX	INCREMENTAL SPACING FOR SCREEN EXAMINATION	
MCUOPT	OPTION TO SUPPRESS INTERMEDIATE OUTPUT WITH MCU STUDIES OPTION (MCUOPT=1 WILL SUPPRESS)	
THE FOLLOWING IDENT RECORDS ARE ALWAYS REQUIRED.		
GO	SIGNIFIES TO BEGIN EXECUTION FOR THIS DATA SET. AFTER EXECUTION, ANOTHER SET OF INPUTS MAY BE READ-IN FOLLOWED BY ANOTHER 'GO' CARD. ANY VALUES ESTABLISHED FROM PREVIOUS INPUT SETS TO THE ROUTINE ARE STILL IN EFFECT. THUS DATA SUCH AS FROM CARD OBSC NEED NOT BE READ AGAIN IF THERE ARE TO BE NO CHANGES TO OBSERVER COORD., ETC.	SMK02060 SMK02070 SMK02080 SMK02090 SMK02100 SMK02110 SMK02120 SMK02130 SMK02140 SMK02150
DONE	*** MUST *** BE THE LAST RECORD READ. DESIGNATES LISTING OF ACCUMULATED EFFECTS IF MORE THAN ONE INPUT SET (DELINEATED BY GO CARDS) WERE INPUT. ALSO RETURNS CONTROL TO EOSAEL EXEC MODULE. NOTE THAT A 'DONE' CARD CAN BE USED IN PLACE OF THE FINAL 'GO' CARD IF DESIRED.	SMK02160 SMK02170 SMK02180 SMK02190 SMK02200 SMK02210 SMK02220 SMK02230 SMK02240 SMK02250
OUTPUTS	LIST OF INPUT PARAMETERS AND AT EACH COMPUTATION TIME, TO, THE FOLLOWING:	SMK02260 SMK02270 SMK02280 SMK02290 SMK02300 SMK02310 SMK02320 SMK02330 SMK02340 SMK02350 SMK02360 SMK02370 SMK02380 SMK02390
	X CLOUD LENGTH ALONG WIND VECTOR (M)	SMK02280
	Z CLOUD LEADING EDGE HEIGHT ALONG WIND VECTOR (M)	SMK02290
YFULL	CLOUD LEADING EDGE FULL-WIDTH PERPENDICULAR TO WIND VECTOR (M)	SMK02300
PATHL	SMOKE LENGTH OF 1 MUNITION ALONG OBS-TGT LOS (M)	SMK02310
CL	TOTAL CL OF SMOKE ALONG OBS-TGT LOS (GM/M**2)	SMK02320
SMTRAN	FOR XN MUNITIONS. TRANSMISSION IN SPECTRAL BANDS OF SMOKE ALONG LOS	SMK02330
SUBROUTINES CALLED... DIRECTLY: CLSMOK, SCONST, SMASSP, JPASCT, WGGEOM, STRANS, BRATE.. INDIRECTLY: XYZINT, QROOT.		
DATA ITTL/ 2HWH,2HIT,2HE ,2HPH,2HOS,2HOR,2HUS,2H ,2HWP,2H >		SMK02400 SMK02410 SMK02420 SMK02430

```

* 2HPW,2HP ,2HOR,2H W,2HP ,2HWI,2HCK,2H/W,2HED,2HGE,2H , SMK024-
* 2HHC,2H S,2HMO,2HKE,2H ,2H ,2H ,2H ,2H ,2H ,2H ,2H ,2H , SMK024-
* 2HFO,2HG ,2HOI,2HL ,2HSM,2HOK,2HE ,2H ,2H ,2H ,2H ,2H , SMK024-
* 2HRE,2HD ,2HPH,2HQS,2HPS,2HOR,2HUS,2H < 2HRP,2H> ,2H ,2H , SMK024-
DATA IR /2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA, SMK02480
*2HRC,2HEX,2HTC,2HG0,2H ,2HB0,2HBN ,2HMI,2HSC,2HOU,2HTP,2H00, SMK02490
*2HNE ,2HNA,2HME /
DATA BRAT1,BRAT2,BRAT3,BRAT4,BRAT5/1,0,0,0,1,/ SMK02500
DATA TYPM,XN,FW,TBURN,EFF,YF,RELHUM,UW,WNDIR,ANGLX,PCAT, SMK02510
*AIRT,TGRAD /13*0./ SMK02520
DATA ITGRAD/0/ SMK02530
BFUNK(T)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.))) SMK02550
**+BRAT5*.07818288*ALOG(1.+358800.*T) SMK02560
NUMDIV=9 SMK02570
NRUNS=0 SMK02580
TIMTRN=1 SMK02590
ICRTR=0 SMK02600
TRANS=1 SMK02610
MCUOPT=0
NDPPNT=0
NUMORE=0
NCY=0
DO 1 J=1,MAXS
CLTOTC(J)=0.
1 CONTINUE
DO 2 J=1,8
2 EXTCC(J)=0.
5 CONTINUE
NRUNS=NRUNS+1
MUNRD=0
KWAVE=0
IF (WAVE1.GE.0.4.AND.WAVE1.LT.0.7) KWAVE=1
IF (WAVE1.GE.0.7.AND.WAVE1.LE.1.2) KWAVE=2
IF (WAVE1.GE.3.0.AND.WAVE1.LE.5.0) KWAVE=4
IF (WAVE1.GE.8.0.AND.WAVE1.LE.12.0) KWAVE=5
IF (WAVE1.GT.1.059.AND.WAVE1.LT.1.061) KWAVE=3
IF (WAVE1.GT.10.59.AND.WAVE1.LT.10.61) KWAVE=6
IF (WAVE1.GT.93.9.AND.WAVE1.LT.94.1) KWAVE=7
IF (WAVE1.GT.3188.AND.WAVE1.LT.3195.) KWAVE=7
IF (KWAVE.EQ.0) GOTO 998
C*** BEGINNING OF READ LOOP
NCHK=0
DO 70 I = 1, 15
    IF(I.EQ.15) GO TO 310
    READ(10IN,20) IR1,IR2,(R1(J),J=1,7)
20   FORMAT(2A2,6X,7F10.3)
C*** RELATING INPUT DATA TO VARIABLE NAMES.
    IF(IR1.EQ.IR(1).AND.IR2.EQ.IR(2)) GOTO 90
    IF(IR1.EQ.IR(3).AND.IR2.EQ.IR(4)) GOTO 100
    IF(IR1.EQ.IR(5).AND.IR2.EQ.IR(6)) GOTO 110
    IF(IR1.EQ.IR(7).AND.IR2.EQ.IR(8)) GOTO 120
    IF(IR1.EQ.IR(9).AND.IR2.EQ.IR(10)) GOTO 130
    IF(IR1.EQ.IR(11).AND.IR2.EQ.IR(12)) GOTO 140
    IF(IR1.EQ.IR(13).AND.IR2.EQ.IR(14)) GOTO 150
    IF(IR1.EQ.IR(15).AND.IR2.EQ.IR(16)) GOTO 155
    IF(IR1.EQ.IR(17).AND.IR2.EQ.IR(18)) GOTO 105
    IF(IR1.EQ.IR(19).AND.IR2.EQ.IR(20)) GOTO 70
    IF(IR1.EQ.IR(21).AND.IR2.EQ.IR(22)) GOTO 115
    IF(IR1.EQ.IR(23).AND.IR2.EQ.IR(24)) GOTO 154
    IF(IR1.EQ.IR(25).AND.IR2.EQ.IR(26)) GOTO 121
    WRITE(10OUT,80)
80   FORMAT(1H ,72HINVALID DATA CARD-DOES NOT CONFORM TO PROPER COSM
*NVENTION IN SMOKE ROUTINE)
    WRITE(10OUT,30) IR1,IR2,(R1(J),J=1,7)
30   FORMAT(1H ,2A2,6X,7F10.3)
        GO TO 999
90   IF (ICLMAT.EQ.1) GOTO 92
    IF (R1(1).NE.0.) RELHUM= R1(1)
    IF (R1(2).NE.0.) UW = R1(2)
    IF (R1(3).NE.0..OR.R1(2).NE.0.) WNDIR = R1(3)

```

```

        IF (R1(4).NE.0.) PCAT = R1(4)
        ICAT=IFIX(PCAT+.0001)
        AIRT = R1(5)
        TGRAD = R1(6)
        ITGRAD=1
        NCHK=1
        GO TO 70
92      IF (R1(1).NE.0.) XN = R1(1)
        IF (R1(2).NE.0.) FW = R1(2)
        IF (R1(3).NE.0.) TBURN = R1(3)
        IF (R1(4).NE.0.) ITYPE = IFIX(R1(4)+.0001)
        IF (MUNRD.EQ.0.OR.R1(5).NE.0.) EFF = R1(5)
        YF=R1(6)
        MUNRD=1
        NCHK=1
        GO TO 70
100     TYPM=R1(1)
        CALL BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,BRAT1,
        *BRAT2,BRAT3,BRAT4,BRAT5)
        IF (IERR.NE.0) WRITE (IOOUT,95) TYPM
95      FORMAT(3PH IN SMOKE, ILLEGAL MUNITION TYPE READ ,F5.0)
        IF (IERR.NE.0) GOTO 999
        IF (R1(2).EQ.0..AND.R1(3).EQ.0..AND.R1(5).EQ.0.) GOTO 93
        *          BRAT1=R1(2)
        BRAT2=R1(3)
        BRAT3=R1(4)
        BRAT4=R1(5)
        BRAT5=R1(6)
93      IF (TYPM.GT.0.) MUNRD=1
        NCHK=1
        GO TO 70
110     ISTO = IFIX(R1(1)+.0001)
        IETO = IFIX(R1(2)+.0001)
        IDTO = IFIX(R1(3)+.0001)
        ANGLX=R1(4)
        NCHK=1
        GO TO 70
115     NOPRNT=0
        IF (R1(1).NE.0.) NOPRNT=1
        CRITER=R1(2)
        ICRTR=IFIX(CRITER+.001)
        IF (ICRTR.GT.2) ICRTR=2
        IF (ICRTR.LT.0) ICRTR=0
        IF (ICRTR.EQ.2) TIMTRN=R1(3)
        IPLT=IFIX(R1(4))
        GOTO 70
120     XM = R1(1)
        YM = R1(2)
        ZM = R1(3)
        TM = R1(4)
        NCHK=1
        GO TO 70
121     MODE=1
        STIME =R1(1)
        ISRN =IFIX(STIME+0.0001)
        FRONT =R1(2)
        DELX =R1(3)
        IF(DELX.LE.0.0) DELX=5.0
        XXX=FRONT/DELX
        NPTS=IFIX(XXX)+1
        MCUOPT=IFIX(R1(4))
        GO TO 70
130     X0 = R1(1)
        Y0 = R1(2)
        Z0 = R1(3)
        NCHK=1
        GO TO 70
140     XT = R1(1)
        YT = R1(2)

```

SMK03130
SMK03140
SMK03150
SMK03160
SMK03170
SMK03180
SMK03190
SMK03200
SMK03210
SMK03220
SMK03230
SMK03240
SMK03250
SMK03260
SMK03270
SMK03280
SMK03290
SMK03300
SMK03310
SMK03320
SMK03330
SMK03340
SMK03350
SMK03360
SMK03370
SMK03380
SMK03390
SMK03400
SMK03410
SMK03420
SMK03430
SMK03440
SMK03450
SMK03460
SMK03470
SMK03480
SMK03490
SMK03500
SMK03510
SMK03520
SMK03530
SMK03540
SMK03550
SMK03560
SMK03570
SMK03580
SMK03590
SMK03600
SMK03610
SMK03620
SMK03630
SMK03640
SMK03650
SMK03660
SMK03670
SMK03680
SMK03690
SMK03700

```

      ZT      = R1(3)          SMK03710
      NCHK=1          GO TO 70          SMK03710
150  DO 152 J=1,7          SMK03710
152  EXTC(J)=R1(J+1)          SMK03740
      EXTC(8)=0,          SMK03750
    70  CONTINUE          SMK03760
154  NMORE=1          SMK03770
      IF (NCHK.EQ.0) NRUNS=NRUNS-1          SMK03780
      IF (NCHK.EQ.0) GOTO 980          SMK03800
155  CONTINUE          SMK03820
C****REDEFINE MUNITION EFFICIENCY IF INPUT AS ZERO          SMK03830
      IF (EFF.GT.0.0) GO TO 11          SMK03840
      IF (ITYPE.EQ.1)EFF=100          SMK03850
      IF (ITYPE.EQ.2)EFF=65.0          SMK03860
      IF (ITYPE.EQ.3)EFF=40.0          SMK03870
      IF (ITYPE.EQ.4)EFF=100.0          SMK03880
      IF (ITYPE.EQ.5)EFF=50.          SMK03890
    11  CONTINUE          SMK03900
      IF (ICLMAT.NE.1) GOTO 12          SMK03910
      RELHUM=RH          SMK03920
      UW   =WNDVEL          SMK03930
      WNDIR=WNDDIR          SMK03940
      ICAT =IPASCT          SMK03950
      PCAT=FLOAT(ICAT)          SMK03960
      AIRT =TEMP          SMK03970
    12  CONTINUE          SMK03980
      IF (IGEOSW.NE.1) GOTO 13
      DISKTM=1000.
C...  CONVERT UNITS FROM KM TO M.          SMK03990
      XT=PTSC(1)*DISKTM          SMK04010
      YT=PTSC(2)*DISKTM          SMK04010
      ZT=PTSC(3)*DISKTM          SMK04020
      XO=PTSC(4)*DISKTM          SMK04030
      YO=PTSC(5)*DISKTM          SMK04040
      ZO=PTSC(6)*DISKTM          SMK04050
      ANGLX=90.          SMK04060
    13  CONTINUE          SMK04070
      IF (ITYPE.LT.1.OR.ITYPE.GT.5) IERR=1          SMK04080
      IF (IERR.EQ.1) WRITE (IOOUT,180) ITYPE          SMK04090
    180  FORMAT(1X,31HIN SMOKE, INVALID SMOKE TYPE = ,I4)
      IF (IERR.EQ.1) GOTO 999          SMK04100
C***  CHECK BURN RATE FOR 100 PERCENT BURN AT TBURN...
      IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0..AND.BRAT5.EQ.0..)          SMK04110
      *BRAT1=1.          SMK04120
      VNORM=BFUNC(1.)
      BRAT1=BRAT1/VNORM          SMK04130
      BRAT2=BRAT2/VNORM          SMK04140
      BRAT3=BRAT3/VNORM          SMK04150
      BRAT4=BRAT4/VNORM          SMK04160
      BRAT5=BRAT5/VNORM          SMK04170
      IF (XN.LE.0.) XN=1.          SMK04180
      IF (ITYPE.EQ.1.AND.TBURN.GT.1.) TBURN=1.          SMK04210
C***  SET UP EXTINCTION COEFF TO BE USED...
      CALL STRANSCL,SMTRAN,ITYPE,EXTC,0)          SMK04220
      CALL SMASSP(XN,EFF,FW,RELHUM, W,ITYPE,YF)
      TGR=TGRAD
      TGRAD=ABS(TGRAD)          SMK04250
      IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 255          SMK04260
      IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 255          SMK04270
      IF (NRUNS.GT.1) WRITE (IOOUT,172) NRUNS          SMK04280
      IF (NRUNS.EQ.1) WRITE (IOOUT,1720) NRUNS          SMK04290
    172  FORMAT(1H*,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,          SMK04300
      *1H*,15X,9HEXECUTION,I3/51X,1H*,15X,1H*/51X,17(1H*)/)          SMK04310
    1720 FORMAT(1H0,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,          SMK04320
      *1H*,15X,9HEXECUTION,I3/51X,1H*,15X,1H*/51X,17(1H*)/)          SMK04330
C***  REPORTING INPUT DATA.
      IF (ICAT.GE.5.AND.ITYPE.NE.4.AND.ITGRAD.EQ.0) WRITE (IOOUT,98) TGRAD          SMK04340
    98  FORMAT(1X,44HIN SMOKE ROUTINE PASQUILL CATEGORIES E AND F,          SMK04350
      *29H REQUIRE TEMPERATURE GRADIENT/10X,23HIF SMOKE IS EXOTHERMIC.,          SMK04360

```

```

*20H A VALUE OF TGRAD = ,F7.2,25H C DEG/M WILL BE ASSUMED./> SMK04380
170 WRITE( IOOUT, 170) SMK04390
FORMAT(5X,15HSMOKE MUNITIONS,22X,25HMETEOROLOGICAL CONDITIONS,25X,SMK04400
123HEXTINCTION COEFFICIENTS) SMK04410
170 WRITE( IOOUT, 190) <ITTL(J,ITYPE),J=1,11>,UW,EXTC(1) SMK04420
190 FORMAT(3X,11A2,11X,10HWINDSPEED ,14X,F5.1,2X,3HM/S,16X, SMK04430
120H0.4-0.7 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04440
170 WRITE( IOOUT, 200) XN,EXTC(2) SMK04450
200 FORMAT(3X,10HNO. ROUNDS,1X,F5.0,17X,22HWIND DIRECTION (USUAL ,28X,SMK04460
120H0.7-1.2 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04470
170 IF (ITYPE.NE.4) WRITE( IOOUT,210) FW,WNDIR,EXTC(3) SMK04480
210 FORMAT(3X,1HFILL WEIGHT,F8.3,3H LB,11X,23HMET CONVENTION AZIMUTH)SMK04490
1,F6.1,2X,7HDEGREES,12X,20H1.06 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04500
170 IF (ITYPE.EQ.4) WRITE( IOOUT,215) FW,WNDIR,EXTC(3) SMK04510
215 FORMAT(3X,9HBURN RATE,2X,F6.1,2X,6HGAL/HR, 8X,23HMET CONVENTION AZSMK04520
1IMUTH) ,F6.1,2X,7HDEGREES,12X,20H1.06 MICROMETERS ,F7.3,2X, SMK04530
27HM**2/GM) SMK04540
170 WRITE( IOOUT,220)TBURN,RELHUM,EXTC(4) SMK04550
220 FORMAT(3X,9HBURN TIME,F8.1,2X,3HSEC,11X,17HRELATIVE HUMIDITY,7X, SMK04560
1F5.1,2X,7HPERCENT,12X,20H3.0-5.0 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04570
1LCAT=JPASCT(ICAT) SMK04580
170 WRITE( IOOUT,230) EFF,LCAT,EXTC(5) SMK04590
230 FORMAT(3X,10HEFFICIENCY,1X,F6.1,2X,7HPERCENT,7X,17HPASQUILL CATEGOSMK04600
1TRY,9X,A1,23X,20H8.0-12. MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04610
170 WRITE( IOOUT,235) YF,AIRT,EXTC(6) SMK04620
235 FORMAT(3X,12HYIELD FACTOR,F6.2,15X,15HAIR TEMPERATURE,8X,F6.1,2X, SMK04630
18HDEGREE C,11X,20H10.6 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04640
170 WRITE( IOOUT,240)TGRAD,EXTC(?) SMK04650
240 FORMAT(36X,14HTEMP GRADIENT,10X,F6.2,9H C DEG./M,11X,
111H94.0 GHZ,9X,F7.3,2X,7HM**2/GM) SMK04660
170 IF (ICAT.GE.5.AND.TGR.LT.0.) WRITE( IOOUT,250) SMK04680
250 FORMAT(36X,24H(ASSUMED POSITIVE INPUT)) SMK04690
170 NRAT=0 SMK04700
255 IF (BRAT1.NE.0.) NRAT=1 SMK04710
IF (BRAT2.NE.0.) NRAT=2 SMK04720
IF (BRAT3.NE.0.) NRAT=3 SMK04730
IF (BRAT4.NE.0.) NRAT=4 SMK04740
170 IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 258 SMK04750
IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 258
IF (NRAT.EQ.1) WRITE( IOOUT,175) BRAT1 SMK04760
170 IF (NRAT.EQ.2) WRITE( IOOUT,176) BRAT1,BRAT2 SMK04770
IF (NRAT.EQ.3) WRITE( IOOUT,177) BRAT1,BRAT2,BRAT3 SMK04780
IF (NRAT.EQ.4) WRITE( IOOUT,178) BRAT1,BRAT2,BRAT3,BRAT4 SMK04790
170 IF (NRAT.EQ.0) WRITE( IOOUT,181) SMK04800
181 FORMAT(40X,19HBURN RATE PROFILE *) SMK04810
170 IF (BRAT5.NE.0.) NRAT=5 SMK04820
NPWP=0 SMK04830
170 IF ((ITYPE.EQ.2,OR,ITYPE.EQ.5),AND,NRAT.GT.1) NPWP=1 SMK04840
170 IF (NRAT.EQ.5) WRITE( IOOUT,179) BRATS SMK04850
179 FORMAT(40X,2H+,F8.4,41H*0.0781829*(358800./<1.+358800.*T/TBURN>))SMK04860
175 FORMAT(1H0,40X,19HBURN RATE PROFILE =,F8.4) SMK04870
176 FORMAT(1H0,31X,19HBURN RATE PROFILE =,F8.4,2H +, SMK04880
*F8.4,10H <T/TBURN>) SMK04890
177 FORMAT(1H0,20X,19HBURN RATE PROFILE =,F8.4,2H +, SMK04900
*F8.4,12H <T/TBURN> +,F8.4,13H <T/TBURN>**2) SMK04910
178 FORMAT(1H0, 9X,19HBURN RATE PROFILE =,F8.4,2H +, SMK04920
*F8.4,12H <T/TBURN> +,F8.4,15H <T/TBURN>**2 +,F8.4, SMK04930
*13H <T/TBURN>**3) SMK04940
C*** PROVIDE COORDINATES.. SMK04950
258 ANGL=ANGLX+180.-WNDIR SMK04960
170 IF (ANGL.GT.360.) ANGL=ANGL-360. SMK04970
IF (ANGL.LT.0. ) ANGL=ANGL+360. SMK04980
170 IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 285 SMK04990
XP1=X0-XM SMK05000
XP2=XT-XM SMK05010
YP1=Y0-YM SMK05020
YP2=YT-YM SMK05030
ZPP1=Z0-ZM SMK05040
ZPP2=ZT-ZM SMK05050
CA=COS(ANGL*PIRAD) SMK05060

```

```

SA=SIN(ANGL*PIRAD)
XPP3=0, SMK05070
YPP3=0, SMK05076
ZPP3=0, SMK05077
XPP1=XP1*CA+YP1*SA SMK05100
YPP1=YP1*CA-XP1*SA SMK05110
XPP2=XP2*CA+YP2*SA SMK05120
YPP2=YP2*CA-XP2*SA SMK05130
IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GO TO 285 SMK0514
260 WRITE (IOOUT,260) XM,YM,ZM,XPP3,YPP3,ZPP3,X0,Y0,Z0,XPP1,YPP1, SMK05150
      1ZPP1,XT,YT,ZT,XPP2,YPP2,ZPP2 SMK05160
      1FORMAT(1H0,30X,17HFIELD COORDINATES,20X,43HROTATED COORD.(WIND X-AS
      1XIS,MUNITION ORIGIN)/27X,3H(X),8X,3H(Y),8X,3H(Z),20X,4H(XW),
      27X,4H(YW),7X,4H(ZW)) SMK05170
      1FORMAT(1X,22HMUNITION COORDINATES= ,3(F9.2,2X),6HMETERS,6X,
      13(F9.2,2X),6HMETERS/1X,22HOBSERVER COORDINATES= ,3(F9.2,2X),
      26HMETERS,6X,3(F9.2,2X),6HMETERS/1X,22HTARGET COORDINATES= ,
      33(F9.2,2X),6HMETERS,6X,3(F9.2,2X),6HMETERS) SMK05180
      1FORMAT(1X,40HANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = , SMK05190
      *F7.2,5H DEG.,/,.5X,12HEVENT TIME = ,F6.1,4H SEC) SMK05200
      IF(MODE.GT.0) WRITE(IOOUT,1000) MODE,STIME,FRONT,DELX SMK05210
      1000 FORMAT(5X,5HMODE=,I2,/,5X,12HSCREEN TIME=,F6.1,/,5X,6HFRONT=,F6.1,
      */,5X,1HINCREMENT=,F6.1) SMK05220
      IF (NOPRNT.EQ.1.OR.MCUOPT.EQ.1) GO TO 285 SMK05230
      IF(MODE.EQ.0) WRITE(IOOUT,280) SMK05240
      IF(MODE.GT.0) WRITE(IOOUT,281) SMK05250
      280 FORMAT(1H0,6X,4HTIME,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,
      110HPATHLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,.7X,5H(SEC),3(1X,8H(METERS
      2ERS)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETERS
      3ERS),/,60X,39H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12., SMK05260
      415H 10.6 94.GHZ/) SMK05270
      281 FORMAT(1H0,6X,5H*L0S*,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,10HPAT
      1HLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,.4X,8H(METERS),3(1X,8H(METERS
      2)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETERS
      3),/,60X,40H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12.,
      415H 10.6 94.GHZ/) SMK05280
      C*** BEGINNING OF CALCULATIONS. SMK05290
      285 IF (W.EQ.0) GOTO 999 SMK05300
      IF (ICAT.LT.1.OR.ICAT.GT.6) GOTO 999 SMK05310
      IF (IDTO.EQ.0) IDTO=1 SMK05320
      IF (ISTO.LE.0) ISTO=IDTO SMK05330
      IF (IETO.LE.ISTO) IETO=ISTO SMK05340
      C*** CALCULATIONS DEPENDENT ON TIME. SMK05350
      C*** SET COMPUTATION TIME STEP DTIME TO 1 SEC. FOR HC/FOG OIL/PWP/
      C*** RP AND WP WICKS/WEDGES SMK05360
      C*** BUT TO TABLE REPORT TIME INCREMENT FOR WP SMK05370
      DTIME=FLOAT(IDTO) SMK05380
      TRANS=1.0 SMK05390
      TSUB(1)=0. SMK05400
      TSUB(2)=TBURN SMK05410
      IF (NPWP.EQ.0) GOTO 620 SMK05420
      IS=IFIX(.5*TBURN+.0001) SMK05430
      IF (IS.LT.1) GOTO 610 SMK05440
      DO 286 I=2,NUMDIV SMK05450
      TSUB(I)=1./5. SMK05460
      286 CONTINUE SMK05470
      DO 600 I=1,IS SMK05480
      FI=FLOAT(I)/5. SMK05490
      T=FI/TBURN SMK05500
      TMS=BFUN(T) SMK05510
      DO 601 JDIV=2,NUMDIV SMK05520
      IF (TMS.LE.QDIV(JDIV)) TSUB(JDIV)=FI SMK05530
      601 CONTINUE SMK05540
      TSUB(NUMDIV)=TBURN SMK05550
      600 CONTINUE SMK05560
      GOTO 620 SMK05570
      610 NPWP=0 SMK05580
      620 CONTINUE SMK05590

```

```

IF (ITYPE.GE.2) DTIME=1. SMK05660
NDIV=NUMDIV-1 SMK05670
DO 700 JPWP=1,NDIV SMK05680
IF (JPWP.GT.1.AND.NPWP.EQ.0) GOTO 700 SMK05690
TSTAGE=TSUB(JPWP) SMK05700
TBRN=TSUB(JPWP+1)-TSTAGE SMK05710
PCNT=QDIV(JPWP+1)-QDIV(JPWP) SMK05720
IF (NPWP.EQ.0) PCNT=1 SMK05730
IF (TBRN.LE.0.) GOTO 700 SMK05740
CALL SCONST(ICAT,UW,ITYPE,FW,EFF,TBRN,NPWP,AIRT,TGRAD,C1,C2,C3, SMK05750
*XN,XLIM,YLIM,ZLIM,HLIM,TLIM,XS,CNEUT,PCNT,W) SMK05760
PCT=PCNT*100 SMK05770
IF (NPWP.NE.0.AND.NOPRNT.NE.1) WRITE (IOOUT,705) PCT,TSUB(JPWP), SMK05780
*TSUB(JPWP+1) SMK05790
IF (NPWP.NE.0.AND.MCUOPT.NE.1) WRITE (IOOUT,705) PCT,TSUB(JPWP),
*TSUB(JPWP+1)
705 FORMAT(1X,3H *** CLOUD PORTION CONTAINING ,F7.3,35H PERCENT OF SMK05800
*DUE DURING BURN FROM ,F7.2,3H TO,F7.2,4H SEC//) SMK05810
CL=0. SMK05820
ICODE=0 SMK05830
TO=0 SMK05840
NCY=0 SMK05850
ICALL=0 SMK05860
ISTT=ISRN-IFIX(TM+0.0001)
II=IFIX(TM)
XPP0=XPP1
I1=IST0
I2=IETO
I3=IDTO
NNN=1
IF(MODE.EQ.0) GO TO 2000
I1=ISTT
I2=ISTT
I3=1
II=0
NNN=NPTS
TO=0
2000 CONTINUE
DU 6 J=1,NNN
IF (ITYPE.EQ.1) TO=FLOAT(I1)-DTIME SMK05870
DO 6 I=I1,I2,I3 SMK05880
L=I-II
IF(MODE.EQ.1) XPP1=XPP0+(J-1)*DELX
IF(MODE.EQ.1) XPP2=XPP1
X=0.0
Y=0.0
Z=0.0
PATHL=0.0
CL=0.0
IF(L.LT.I1) GO TO 2001
C*** TAB IS NEXT TABLE REPORT TIME. TO IS NEXT COMPUTATION TIME. SMK05890
TAB=FLOAT(L)-TSTAGE SMK05900
IF (TAB.GT.0.) GOTO 3 SMK05910
IF (NCY.GE.MAXS) GOTO 6 SMK05920
NCY=NCY+1 SMK05930
GOTO 6 SMK05940
3 TO=TO+DTIME SMK05950
CALL WGGEOM(ICALL,CLGAUS,ITYPE,XPP1,YPP1,ZPP1,XPP2,YPP2,ZPP2,C1, SMK05960
C2,C3,TO,UW,ICAT,HLIM,TLIM,CNEUT,XS,PATHL,X,Y,Z,XLIM,YLIM,ZLIM) SMK05970
CALL CLSMOK(ICODE,CLGAUS,ITYPE,CL,W,PATHL,TBURN,TBRN,PCNT,TSTAGE, SMK05980
*NPWP,XLIM,YLIM,ZLIM,TO,TLIM,DTIME,X,Y,Z,BRAT1,BRAT2,BRAT3,BRAT4, SMK05990
*BRAT5) SMK06000
C*** REPORT OUTPUT DATA IF TIME TO .EQ. TAB, OTHERWISE, LOOP SMK06010
C*** BACK FOR NEXT TIME STEP. SMK06020
IF (TO.LT.(TAB-.001)) GOTO 3 SMK06030
2001 CALL STRANSK(CL,SMTRAN,ITYPE,EXTC,1) SMK06040
YFULL=2.*Y SMK06050
TV=FLOAT(I)-TM SMK06060
XWRIT=FLOAT(I)
IF(MODE.GT.0) XWRIT=XPP2+XM

```

```

PRNTOP=NOPRNT
IF (MODE.GT.0) PRNTOP=MCUOPT
IF (PRNTOP.NE.1) WRITE(IODOUT,290) XWRIT,X,YFULL,Z,PATHL,CL,           SMK06070
  *(SMTRANCK),K=1,7)
290 FORMAT(6X,F5.0,2X,F6.0,3X,F6.0,3X,F6.0,4X,F6.2,4X,F7.2,7(3X,F5.3))SMK06090
  IF (ICRTR.EQ.2.AND.TV.GT.(TIMTRN+.5)) GOTO ?                      SMK06100
  IF (ICRTR.EQ.1.AND.SMTRAN(KWAVE).GE.TRANS) GOTO ?                  SMK06110
  TRTM=TV
  TRANS=SMTRAN(KWAVE)
?
CONTINUE
IF (NCY.GE.MAXS) GOTO 6                                              SMK06130
NCY=NCY+1
CLTOT(NCY)=CLTOT(NCY)+CL
6 CONTINUE
700 CONTINUE
C*** FINAL OUTPUT
IF (NOMORE.EQ.0) GOTO 5                                              SMK06190
980 IF (NCY.LE.0) GOTO 999
  IF (NRUNS.LE.1.AND.NPWF.EQ.0) GOTO 997                           SMK06200
  IF(MODE.EQ.0) WRITE(IODOUT,982) NRUNS                               SMK06220
  IF(MODE.GT.0) WRITE(IODOUT,984) NRUNS                               SMK06240
982 FORMAT(1H1,40X,19HCOMBINED EFFECT OF,13,21H EXECUTIONS IN SMOKE:/SMK06250
  *10X,10HTIME (SEC),5X,11HCL (G/M**2),21X,12HTRANSMISSION/          SMK06260
  *10X,10(1H-),5X,11(1H-),4X,                                         SMK06270
  *54H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12. 10.6 94.GHZ/)           SMK06280
984 FORMAT(1H1,40X,19HCOMBINED EFFECT OF,13,21H EXECUTIONS IN SMOKE:/,
  *10X,11HL0S(METERS),4X,11HCL (G/M**2),21X,12HTRANSMISSION/
  *10X,10(1H-),5X,11(1H-),4X,                                         SMK06290
  *54H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12. 10.6 94.GHZ/)           SMK06300
  TRANS=1.0
  DO 985 I=1,NCY
  CL=CLTOT(I)
  TU=FLOAT(IST0+(I-1)*IDTO)
  CALL STRANS(CL,SMTRAN,ITYPE,EXTC,1)                                SMK06310
  YWRIT=TO
  IF(MODE.GT.0) YWRIT=XPP0+XM+(I-1)*DELX
  WRITE(IODOUT,983) YWRIT,CL,(SMTRAN(J),J=1,7)                         SMK06320
  IF(IPLT.EQ.1) WRITE(NPLOTU,883) YWRIT,CL
883 FORMAT(1X,F6.1,2X,F8.3)                                               SMK06350
983 FORMAT(12X,F6.0,8X,F8.2,6X,7(1X,F5.3,2X))                          SMK06360
  IF (ICRTR.EQ.2.AND.TO.GT.(TIMTRN+.5)) GOTO 985                     SMK06370
  IF (ICRTR.EQ.1.AND.SMTRAN(KWAVE).GE.TRANS) GOTO 985                  SMK06380
  TRTM=TO
  TRANS=SMTRAN(KWAVE)
985 CONTINUE
997 CONTINUE
WRITE(IODOUT,3100)WAVE1,TRANS,TRTM
TRANS=SMTRAN(KWAVE)
RETURN
998 WRITE(IODOUT,3200)
GOTO 999
3100 FORMAT(1H0,5X,37H***TRANSMISSION RETURNED TO MAIN FOR
1        14HWAVELENGTH OF ,F8.3,16H MICROMETERS IS ,F5.3,
2        8H AT TIME ,F7.0)                                              SMK06480
3200 FORMAT(1H0,10X,35HINVALID WAVELENGTH PASSED FROM MAIN
1        ,/,10X,27H TRANS=1.0 RETURNED TO MAIN,//)                      SMK06510
310 CONTINUE
C*** ERROR CONDITION
WRITE(IODOUT,320)
320 FORMAT(1H,105HMORE THAN 13 DATA CARDS HAVE BEEN INPUT. PLEASE CHSMK06550
  *ECK THAT THERE ARE NO MORE THAN 13 DATA CARDS PER RUN.)
999 CONTINUE
IERR=1
TRANS=1.0
RETURN
END

```

```

SUBROUTINE CLSMOK(ICODE,CLGAUS,ITYPE,CL,W,PATHL,TBURN,TBRN,PCNT, CLS00010
*TSTAGE,NPWP,XLIM,YLIM,ZLIM,TO,TLIM,DTIME,X,Y,Z,BRAT1,BRAT2, CLS00020
*BRAT3,BRAT4,BRAT5) CLS00030
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK CLS00040
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIR TU,NCIMT,KSTOR,NPLOTU CLS00050
CLS00060
C NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION CLS00070
AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC) CLS00080
CLS00090
COMMON /M05/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9), CLS00100
*EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2) CLS00110
CLS00120
NOTE: TO CHANGE MAXIMUM BURN OR OBSCURATION DURATION, CHANGE CLS00130
PARAMETER MAXS AND DIMENSIONS OF SMAS,PVOL,CLTOT TO MAXS CLS00140
CLS00150
THIS SUBROUTINE CALCULATES THE VOLUME OF THE SMOKE CLOUD AND CLS00160
THE CONCENTRATION * LENGTH(CL) AT THE INTERSECTION OF THE OBSERVER CLS00170
TARGET LGS OF THE CLOUD FORMED BY THE TOTAL NUMBER OF SMOKE CLS00180
MUNITIONS WHICH WERE DETONATED AT THE SAME TIME AND FROM THE SAME CLS00190
POINT. FOR CONTINUOUS TYPE BURNS (HC, FOG OIL, PWP, RP, AND CLS00200
WP WICKS OR WEDGES), THE BURN MASS INCREMENT AND PATH LENGTH CLS00210
TO VOLUME RATIO ARE STORED FOR EACH PUFF. THESE PUFFS ARE CLS00220
SUBSEQUENTLY ADDED TO FIND THE TOTAL EFFECT OF OBSCURANT. CLS00230
CLS00240
CLS00250
INPUTS CLS00260
CLS00270
CLS00280
CLS00290
CLS00300
CLS00310
CLS00320
CLS00330
CLS00340
CLS00350
CLS00360
CLS00370
CLS00380
CLS00390
CLS00400
CLS00410
CLS00420
CLS00430
CLS00440
CLS00450
CLS00460
CLS00470
CLS00480
CLS00490
CLS00500
CLS00510
CLS00520
CLS00530
CLS00540
CLS00550
CLS00560
CLS00570
CLS00580
CLS00590
CLS00600
CLS00610
CLS00620
CLS00630
CLS00640
CLS00650
CLS00660
CLS00670
CLS00680
CLS00690
CLS00700
OUTPUTS CLS00490
CLS00500
CLS00510
CLS00520
CLS00530
CLS00540
CLS00550
CLS00560
CLS00570
CLS00580
CLS00590
CLS00600
CLS00610
CLS00620
CLS00630
CLS00640
CLS00650
CLS00660
CLS00670
CLS00680
CLS00690
CLS00700
C*** SIMPLE FOR INSTANTANEOUS BURN OF WP CLS00590
BFUN(T)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.))) +BRAT5* CLS00600
*.07818288*ALOG(1.+358800.*T) CLS00610
MAXS=960 CLS00620
IF (ITYPE.GT.1) GOTO 100 CLS00630
VOL=0.25*(4.*PI/3.)**X*Y*Z CLS00640
CL=W*(.25*PATHL/VOL+.75*CLGAUS) CLS00650
RETURN CLS00660
C*** INITIALIZE CLS00670
100 CL=0 CLS00680
IF (ICODE.NE.0) GOTO 110 CLS00690
ICODE=1 CLS00700
IB=0
IPL=0
START=0
BRNDOUT=FLOAT(IFIX(.9999+TBRN))
NRAT=2
IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0..) NRAT=1
IF (BRAT5.NE.0.) NRAT=5

```

```

KSUB=0          CLS00710
TOTPV=0         CLS00720
BR=W           CLS00730
IF (<ITYPE,NE,4>) BR=W/BRNOUT CLS00740
IF (<NPWP,NE,0>) BR=BR*PCNT CLS00750
IF (<BRNOUT,LE,1>) NRAT=1 CLS00760
IF (<NRAT,GT,1>) BTHEN=BFUNK(TSTAGE/TBURN) CLS00770
C*** STORE PUFF MASS EMITTED AT TIME TO CLS00780
110 IF (<TO,GT,BRNOUT>) GOTO 120 CLS00790
IF (<IB,GE,MAXS>) GOTO 120 CLS00800
IB=IB+1         CLS00810
IF (<NRAT,EQ,1>) GOTO 120 CLS00820
IF (<IB,EG,MAXS>) WRITE (1000,900) IB CLS00830
900 FORMAT(1X,61H*** WARNING - IN SMOKE, MAXIMUM STORAGE FOR BURN DURA CLS00840
*TION OF ,15,55H SEC IS FULL. ACCURACY BEYOND THIS POINT DECREASES. CLS00850
* *** > CLS00860
TM=TO          CLS00870
IF (<TO,GT,TBRN>) TM=TBRN CLS00880
T=<(TM+TSTAGE)/TBURN> CLS00890
BNOW=BFUNK(T) CLS00900
SMAS(ib)=W*(BNOW-BTHEN) CLS00910
BTHEN=BNOW CLS00920
IF (<ITYPE,EQ,4>) SMAS(ib)=SMAS(ib)*TBURN CLS00930
C*** COMPUTE VOLUME AT TIME TO OF FIRST PUFF CLS00940
120 IF (<ITYPE,EQ,3>) GOTO 130 CLS00950
IF (<ITYPE,EQ,4>) GOTO 140 CLS00960
C*** PWP, RP OR WP WICKS/WEDGES CLOUD CLS00970
VOL=0.25*<4.*PI/3.>*X*Y*Z CLS00980
GOTO 200       CLS00990
130 IF (<TO,LE,TLIM>) GOTO 140 CLS01000
C*** POST RISE REGION HC CONE CLS01010
VOL=0.5*<PI/3.>*XLIM*YLIM*ZLIM CLS01020
C*** POST-RISE FRUSTRUM OF APPROXIMATED ELLIPTIC CONE. CLS01030
XZPROJ=Z*<(X-XLIM)/(Z-ZLIM)> CLS01040
VFRUST=0.5*<PI/3.>*(XZPROJ*(Y*Z-YLIM*ZLIM)+(X-XLIM)*YLIM*ZLIM) CLS01050
VOL=VOL+VFRUST CLS01060
GOTO 200       CLS01070
C*** HC BEFORE THE END OF EXOTHERMIC RISE AND FOG OIL CASE CLS01080
140 VOL=0.5*<PI/3.>*X*Y*Z CLS01090
C*** STORE PATH LENGTH/VOLUME RATIO AT TIME TO FOR FIRST EXPANDING CLS01100
C UNIFORM AND GAUSSIAN PUFF CONTRIBUTION. CLS01110
200 IF (<ITYPE,NE,4>) PV=.25*(PATHL/VOL)+.75*CLGAUS CLS01120
IF (<ITYPE,EQ,4>) PV=PATHL/VOL CLS01130
IF (<PV,LE,0.,AND,START,EQ,0,>) RETURN CLS01140
IF (<PV,GT,0.,AND,START,EQ,0,>) START=TO CLS01150
IF (<IPL,GE,MAXS>) GOTO 300 CLS01160
IPL=IPL+1      CLS01170
IF (<IPL,EG,MAXS>) WRITE (1000,910) TO,IPL CLS01180
910 FORMAT(1X,58H*** WARNING - IN SMOKE, MAX. STORAGE FOR CLOUD VOLUME CLS01190
*S OF ,15,34H SEC. IS FULL. COMPUTATION TIME = ,F6,0,5H SEC./ CLS01200
*15X,40HACCURACY BEYOND THIS POINT DECREASES ***> CLS01210
PVOL(IPL)=PV CLS01220
C*** SUM CL FOR PUFFS CLS01230
300 IF (<NRAT,EQ,1>) GOTO 400 CLS01240
LMIN=MAX(0,(IPL-IB+1)) CLS01250
IF (<IPL,LT,LMIN>) RETURN CLS01260
N=0             CLS01270
DO 320 J=LMIN,IPL CLS01280
K=IPL-J+1      CLS01290
CDEL=SMAS(K)*PVOL(J) CLS01300
CL=CL+CDEL     CLS01310
N=N+1           CLS01320
IF (<N,LT,120>) GOTO 320 CLS01330
IF (<ABS(CDEL),LT,1.E-5>) RETURN CLS01340
320 CONTINUE      CLS01350
RETURN          CLS01360
C*** FAST COMPUTATION FOR CONSTANT BURN RATES CLS01370
400 TOTPV=TOPV+PVOL(IPL) CLS01380
C INDEX OF LAST PUFF EMITTED CLS01390
IF (<IPL,LT,MAXS>) KSUB=IPL-IB CLS01400

```

```
C KEEP REMOVING LAST PUFF IF POSSIBLE          CLS01410
    IF (<IPL,EQ,MAXS) KSUB=KSUB+1           CLS01420
    IF (<KSUB,GT,IPL) KSUB=IPL             CLS01430
C ONLY REMOVE PUFFS AFTER BURN HAS STOPPED (IE LAST PUFF OUT) CLS01440
    IF (<KSUB,GT,0) TOTPV=TOPV-PVOL(KSUB)   CLS01450
    CL=TOPV*BR                           CLS01460
    RETURN                                CLS01470
    END                                    CLS01480
```

SUBROUTINE SCONST(ICAT,UW,ITYPE,FW,EFF,TBRN,NPWP,AIRT,TGRAD,C1,C2,SC000010
 * C3, XN,XLIM,YLIM,ZLIM,HLIM,TLIM,XS,CNEUT,PCNT,W) SC000020
 COMMON /CONST/PI,PI2,PIRAD,TWOPi,TORRMB,CDEGK SC000030
 COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIR TU,NCLIMT,KSTOR,NPLOTUSC000040
 SMOKE MODEL PARAMETERS FOR CLOUD DIMENSIONS SC000050
 THIS SUBROUTINE CALCULATES THE PARAMETERS FOR THE CLOUD WIDTH AND SC000060
 CLOUD HEIGHT(MOMENTUM) AS A FUNCTION OF PASQUILL CATEGORY AND THE SC000070
 HEAT RISE PARAMETER FOR THE HEIGHT(EXOTHERMIC). HEAT RISE IS SC000080
 TERMINATED AS A FUNCTION OF ATMOSPHERIC STABILITY SC000090
 INPUTS ICAT PASQUILL CATEGORY SC000100
 1-A, 2-B, 3-C, 4-D, 5-E, 6-F SC000110
 UW WIND VELOCITY (M/S) SC000120
 ITYPE TYPE OF SMOKE. 1=WP, 2=PWP OR WP WICK/WEDGE, SC000130
 3=HC, 4=FOG OIL, 5=RP SC000140
 FW FILL WEIGHT (LBS) SC000150
 EFF EFFICIENCY (PERCENT) SC000160
 XN NUMBER OF MUNITIONS SC000170
 TBRN BURN TIME FOR THIS STAGE (OR ENTIRE MUNITION) SEC. SC000180
 NPWP FLAG FOR LONG-BURN PHOSPHORUS. IF NON-ZERO, THEN SC000190
 BURNS ARE IN 25 PERCENT STAGES TO ALLOW SEPARATE SC000200
 BUOYANCIES FOR EACH. SC000210
 AIRT SURFACE AIR TEMPERATURE (DEG C) SC000220
 TGRAD VERT TEMP GRADIENT (C DEG./M) USED ONLY FOR SC000230
 CATEGORIES E,F (IE. 5,6) IN WHICH IT MUST SC000240
 BE POSITIVE SC000250
 OUTPUTS C1 EQUATION PARAMETER FOR CLOUD WIDTH Y=9.1+C1*X SC000260
 C2 PARAMETER FOR HEIGHT (MOM) Z=2.73+C2*X SC000270
 C3 PARAMETER FOR HEIGHT (EX0) Z=2.73+C2*X+C3*X**2/3 SC000280
 XLIM CLOUD LENGTH ALONG WIND DIRECTION (M) SC000290
 AT RISE TERMINATION. SC000300
 YLIM CLOUD BASE HALF-WIDTH PERP. WIND DIRECTION (M) SC000310
 AT RISE TERMINATION. SC000320
 ZLIM TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE. SC000330
 HLIM TOTAL ADDED RISE AT TERMINATION OF HEAT RISE. SC000340
 TLIM TIME OF TERMINATION OF HEAT RISE. SC000350
 XS TERM FOR CALCULATION OF HEIGHT FOR NEUTRAL CONDITIONS SC000360
 CNEUT TERM FOR CALCULATION OF HEIGHT FOR NEUTRAL CONDITIONS SC000370
 SC000380
 SC000390
 DIMENSION C11(6),C22(6) SC000400
 DATA C11/.419/.328,.238,.2,.18,.146/ SC000410
 DATA C22/.137,.11,.073,.066,.055,.046/ SC000420
 TLIM=600. SC000430
 XS=0.0 SC000440
 CNEUT=0.0 SC000450
 XLIM=0. SC000460
 YLIM=0. SC000470
 ZLIM=0. SC000480
 HLIM=1.0 SC000490
 C*** ERROR CONDITION SC000500
 IF(ICAT.LT.1.OR.ICAT.GT.6) WRITE(I0OUT,50) SC000510
 50 FORMAT(1H,62ERROR IN SUBROUTINE SCONST. PASQUILL CATEGORY IS NOT SC000520
 1ACCEPTABLE) SC000530
 C*** IF (ICAT.LT.1.OR.ICAT.GT.6) RETURN SC000540
 C*** SELECTION OF CLOUD WIDTH COEFFICIENT AS A FUNCTION OF PASQUILL SC000550
 C*** CATEGORY. SC000560
 C1 = C11(ICAT) SC000570
 C*** SELECTION OF CLOUD HEIGHT(MOM) COEFFICIENT AS A FUNCTION OF SC000580
 PASQUILL CATEGORY. SC000590
 C2 = C22(ICAT) SC000600
 IF(ICAT.GT.1)GO TO 1 SC000610
 IF(UW.GT.2.0)GO TO 1 SC000620
 C2=.15 SC000630
 IF(UW.GT.1.5)GO TO 1 SC000640
 C2=.18 SC000650
 IF(UW.GT.1.0)GO TO 1 SC000660
 C2=.25 SC000670
 IF(UW.GT.0.5)GO TO 1 SC000680
 C2=.39 SC000690
 1 CONTINUE SC000700

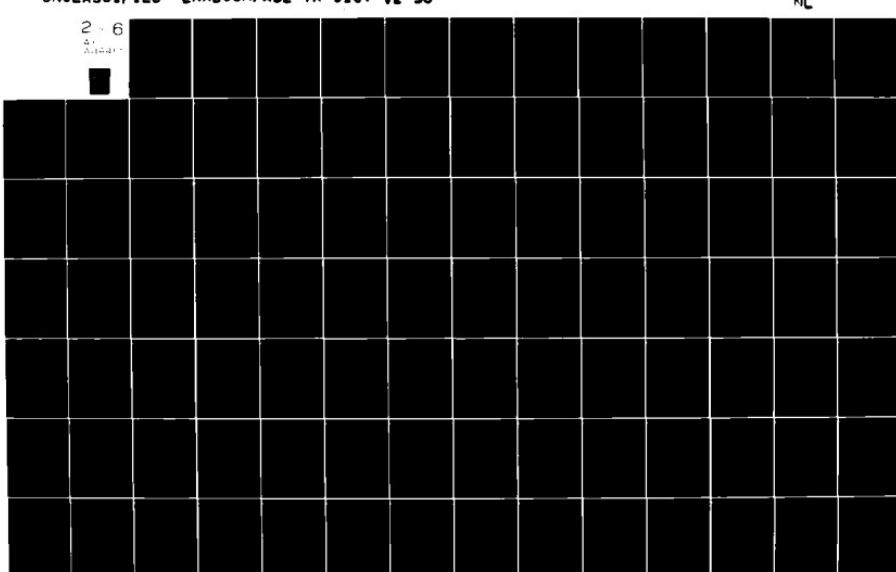
AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS=ETC F/G 11/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(1)
FEB 82 R G STEINHOFF

UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU

NL

2 ~ 6

20
21



```

120 TB=TBRN
    IF (<ITYPE.EQ.1.AND.TBRN.GT.1.>) TB=1.          SC000710
    IF (NPWP.NE.0) TB=TB/PCNT                      SC000720
C*** IF NON-EXOTHERMIC, RETURN.                  SC000730
    IF (<ITYPE.EQ.4>) RETURN                      SC000740
    EM=900.0                                       SC000750
    IF(<ITYPE.EQ.3>)EM=500.0                      SC000760
    IF(<ITYPE.EQ.5>)EM=660.                         SC000770
    F=0.00001*3.59*453.59*EM*FW*XN*(EFF/100.)/TB   SC000780
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT FOR UNSTABLE ATM. SC000800
C*** CONDITIONS (A,B,C)                           SC000810
    C3=1.6/UW*(F**.3333)                          SC000820
    IF(<ITYPE.EQ.3>) GO TO 200                   SC000830
    IF(ICAT.EQ.4) GO TO 131                      SC000840
    IF(ICAT.GT.4) GO TO 141                      SC000850
    TLIM=0.0                                       SC000860
C*** DETERMINE TIME AND TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE. SC000870
123 TLIM=TLIM+2.                                  SC000880
    X=UW*TLIM                                     SC000890
    Y=(9.1+C1*X)/2.                                SC000900
    V=.75*(2.*3.14159)**1.5*(Y/2.15)**2*(2.73+C3*X)/4.3 SC000910
    C=W/(V*TB)                                    SC000920
    IF(C.GT.0.11) GO TO 123                      SC000930
    HLIM=C3*X**.667                               SC000940
    RETURN                                         SC000950
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENTS FOR NEUTRAL ATM. SC000960
C*** CONDITIONS (D).
131 XS=10.0*F**0.4                               SC000970
    CNEUT=1.6*(F**.3333)*XS**0.667/UW           SC000980
    RETURN                                         SC000990
141 CONTINUE                                      SC001000
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT AND TOTAL CLOUD SC001020
C*** HEIGHT AT TERMINATION OF HEAT RISE FOR STABLE ATM. CONDITIONS(E,F) SC001030
    SBAR=9.8/(AIRT+273.0)*(TGRAD+0.0098)        SC001040
    HLIM=1.4*(F/(UW*SBAR))**.333                 SC001050
    RETURN                                         SC001060
C *** CALCULATE PARAMETERS FOR HC 10 SEC. RISE TIME. SC001070
200 TLIM=10.                                     SC001080
    XLIM=TLIM*UW                                 SC001090
    YLIM=(9.1+C1*XLIM)/2.                        SC001100
    ZLIM=2.73+C2*XLIM                           SC001110
    IF (<ICAT.EQ.4>) GOTO 231                  SC001120
    HLIM=C3*XLIM**.667                          SC001130
    ZLIM=ZLIM+HLIM                               SC001140
    IF (<ICAT.LT.5>) RETURN                     SC001150
    SBAR=9.8/(AIRT+273.)**(TGRAD+0.0098)       SC001160
    HTEST=1.4*(F/(UW*SBAR))**.333              SC001170
    IF (<HLIM.LE.HTEST>) RETURN                SC001180
C*** IF UNSTABLE ATMOSPHERE REACHES MAX. BEFORE 10 SEC. COMPUTE SC001190
C*** TLIM.
    XLIM=(HTEST/C3)**1.5                         SC001200
    TLIM=XLIM/UW                                 SC001210
    YLIM=(9.1+C1*XLIM)/2.                        SC001220
    HLIM=HTEST                                    SC001230
    ZLIM=2.73+C2*XLIM+HLIM                      SC001240
    RETURN                                         SC001250
231 XS=10.0*F**0.4                               SC001260
    CNEUT=C3*(XS**.667)                          SC001270
    HLIM=CNEUT*(0.4+0.64*(XLIM/XS)+2.2*(XLIM/XS)**2)/ SC001280
    *(1.+0.8*XLIM/XS)**2                         SC001290
    ZLIM=ZLIM+HLIM                               SC001300
    RETURN                                         SC001310
    END                                            SC001320
                                                SC001330

```



```

SUBROUTINE STRANS(CL,SMTRAN,ITYPE,EXTC,ICALL)

      SMOKE MODEL CLOUD TRANSMISSION
      THIS SUBROUTINE CALCULATES TRANSMISSION IN 7 SPECTRAL REGIONS
      (0.4-0.7, 0.7-1.2, 1.06, 3.0-5.0, 8.0-12.0, 10.6 MICROMETERS AND 94.GHZ)
      FOR A GIVEN SMOKE TYPE AND CONCENTRATION LENGTH
      INPUTS ITYPE SMOKE TYPE
              1 WHITE PHOSPHOROUS
              2 PLASTICIZED WHITE PHOSPHORUS, WP WICK WEDGE
              3 HC
              4 FOG OIL
              5 RED PHOSPHORUS
      CL     COMPUTED CL IN (G/M**2)
      ICALL = 0 SETS UP EXTC ARRAY USED FOR COMPUTATIONS AND
              ALLOWS USER TO OVER-RIDE DEFAULT ALPHAS.
              = 1 EXECUTES TRANSMISSION CALCULATION.

      OUTPUTS SMTRAN TRANSMISSION THROUGH SMOKE (DECIMAL)
      EXTC ARRAY OF EXTINCTION COEFF. ACTUALLY USED
      IN TRANSMISSION CALCULATION. EXTC(8) IS
      USED AS A FLAG FOR ICALL=0 REPLACEMENT.

      IF EXTC(8)=ITYPE, NO CHANGES ARE MADE IN EXTC ARRAY IE.,
      ALPHA VALUES DO NOT REPLACE EXTC VALUES.
      IF EXTC(8)=0, ONLY THOSE VALUES IN EXTC WHICH ARE
      ZERO ARE REPLACED BY THE STORED VALUES IN
      ALPHA (ITYPE COLUMN).
      IF EXTC(8) IS NOT 0, OR ITYPE, THEN ALL EXTC VALUES
      ARE REPLACED BY CORRESPONDING ALPHA VALUES AND
      EXTC(8) IS SET TO ITYPE.

      DIMENSION ALPHA(7,5),SMTRAN(7),EXTC(8)
      DATA ALPHA /4.304,2.166,1.541,0.350,0.338,0.364,0.001,
      *           4.304,2.166,1.541,0.350,0.338,0.364,0.001,
      *           4.579,2.186,2.040,0.190,0.052,0.051,0.001,
      *           6.851,4.592,3.497,0.245,0.020,0.018,0.001,
      *           4.304,2.166,1.541,0.350,0.338,0.364,0.001/
      C*** TRANSMISSION CALCULATED BY BEER'S LAW APPROXIMATION
      C*** IF ICALL=0, EXTC ARRAY IS FORMED OR MODIFIED...
      IF (ICALL.NE.0) GOTO 20
      IF (EXTC(8).EQ.FLOAT(ITYPE)) GOTO 18
      IF (EXTC(8).EQ.0.) GOTO 15
      DO 13 J=1,7
13  EXTC(J)=0
15  EXTC(8)=FLOAT(ITYPE)
      DO 17 J=1,7
      IF (EXTC(J).EQ.0.) EXTC(J)=ALPHA(J,ITYPE)
17  CONTINUE
18  RETURN
      C*** FOR ICALL NON-ZERO, COMPUTE TRANSMISSION USING EXTC VALUES.
      20  DO 30 I=1,7
      SMTRAN(I)=EXP(-EXTC(I)*CL)
30  CONTINUE
      RETURN
      END

```

```

SUBROUTINE WGGEOM( ICALL, CLGAUS, ITYPE, XPP1, YPP1, ZPP1, XPP2, YPP2,
1ZPP2, C1, C2, C3, TO, UW, ICAT, HLIM, TLIM, CNEUT, XS, PATHL, X, Y, Z, XLIM,
2YLIM, ZLIM) WGG00010
WGG00020
WGG00030
WGG00040
WGG00050
WGG00060
WGG00070
WGG00080
WGG00090
WGG00100
WGG00110
WGG00120
WGG00130
WGG00140
WGG00150
WGG00160
WGG00170
WGG00180
WGG00190
WGG00200
WGG00210
WGG00220
WGG00230
WGG00240
WGG00250
WGG00260
WGG00270
WGG00280
WGG00290
WGG00300
WGG00310
WGG00320
WGG00330
WGG00340
WGG00350
WGG00360
WGG00370
WGG00380
WGG00390
WGG00400
WGG00410
WGG00420
WGG00430
WGG00440
WGG00450
WGG00460
WGG00470
WGG00480
WGG00490
WGG00500
WGG00510
WGG00520
WGG00530
WGG00540
WGG00550
WGG00560
WGG00570
WGG00580
WGG00590
WGG00600
WGG00610
WGG00620
WGG00630
WGG00640
WGG00650
WGG00660
WGG00670
WGG00680
WGG00690
WGG00700

```

C NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION
AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)

COMMON /M05/ SMAS(960), PVOL(960), CLTOT(960), SMTRAN(7), R1(9),
*EXT(8), ZL(2), XL(2), YL(2), XINT(2), YINT(2), ZINT(2), IFLAG(2)
COMMON /CONST/ PI, PI2, PIRAD, TWOPI, TORRMB, CDECK

C DATA MAXS /960/

C NOTE: TO CHANGE LENGTH OF BURN OR OBSCURATION, RESET
MAXS AND REDIMENSION SMAS, PVOL AND CLTOT ARRAYS

SMOKE MODEL GEOMETRY

INPUT:
ITYPE = SMOKE TYPE CODE
XPP1, YPP1, ZPP1 = OBSERVER IN MUNITION CENTERED COORDINATES
YPP1, YPP2, ZPP2 = TARGET IN MUNITION CENTERED COORD. (X AXIS ALONG
WIND VECTOR.)
C1, C2, C3 = CLOUD GROWTH PARAMETERS (SEE PROGRAM SCONST)
TO = TIME SINCE DETONATION (SECONDS)
UW = WIND SPEED (METERS/SECOND)
ICAT=PASQUILL CATEGORY
HLIM, TLIM, CNEUT, XS= EXOTHERMIC RISE PARAMETERS
ICALL = FLAG SET BY USER TO 0 FOR 1ST CALL. RESET BY PROGRAM
TO 1 THEREAFTER.

OUTPUT:
X = CLOUD LENGTH (METERS)
Y = CLOUD HALF-WIDTH (METERS)
Z = CLOUD HEIGHT (METERS)
PATHL = PATHLENGTH OF LOS THROUGH SMOKE CLOUD

SUBROUTINES CALLED... DIRECTLY: XYZINT, GPUFF, INDIRECTLY: QROOT

*** TRANSLATION TO PLACE MUNITION AT (0,0,0)
IF (ICALL.NE.0) GOTO 10

ICALL=1
KCALL=0

C*** CALCULATE LEADING EDGE LOCATION AT TO.
10 A=UW*TO
B=(9.1/2.0)+A*C1/2.
IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) GOTO 33

C*** WP/PWP/RP COMPUTATION.
11 GO TO (11,11,11,21,31,31), ICAT
CONTINUE
C=2.73+C2*A+C3*A**.667
CLIM=2.73+C2*A+HLIM
IF(TO.GT.TLIM.AND.C.GT.CLIM)C=CLIM
GO TO 41

21 CONTINUE
C=2.73+C2*A
IF (A.LE.XS) C=C+C3*A**.667
IF (A.GT.XS) C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/
1(1.0+0.8*(A/XS))**2
GO TO 41

31 CONTINUE
C=2.73+C2*A+C3*A**0.667
CLIM=2.73+C2*A+HLIM
IF(C.GT.CLIM)C=CLIM
GOTO 41

C*** HC, FOG OIL COMPUTATION.
33 C=2.73+C2*A
IF (ITYPE.EQ.4) GOTO 41
IF (TO.LT.TLIM) GOTO 35
C=C+HLIM
GOTO 41

35 IF (ICAT.NE.4.OR.A.LE.XS)C=C+C3*A**0.667
IF (ICAT.EQ.4 .AND. A.GT.XS)

```

    I C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/
    1<1.0+0.8*(A/XS)**2>***2          WGG00710
    41 CONTINUE                         WGG00720
C*** NOW COMPUTE PATH LENGTHS -- SELECT ELLIPSOID FOR PHOSPHORUS   WGG00730
C*** OR ELLIPTICAL CONES FOR HC, FOG OIL                           WGG00740
C*** FIRST GAUSSIAN PUFF AND IMAGE PUFF CONTRIBUTION, THEN FULLY MIXED WGG00750
    NCODE=2                           WGG00760
    IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) NCODE=1                         WGG00770
    IF (ITYPE.NE.4)                  WGG00780
    *CALL GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1,ZPP2) WGG00800
    X=A                             WGG00810
    Y=B                             WGG00820
    Z=C                             WGG00830
    IF (ITYPE.EQ.3) GOTO 200           WGG00840
100   X0=A                           WGG00850
    Y0=B                           WGG00860
    Z0=C                           WGG00870
150   XL<1>=XPP1                     WGG00880
    XL<2>=XPP2                     WGG00890
    YL<1>=YPP1                     WGG00900
    YL<2>=YPP2                     WGG00910
    ZL<1>=ZPP1                     WGG00920
    ZL<2>=ZPP2                     WGG00930
    CALL XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)      WGG00940
    PATHL=SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+        WGG00950
    *(ZINT(1)-ZINT(2))**2)
    IF (ITYPE.EQ.3) GOTO 220         WGG00960
    RETURN                          WGG00970
200   ICONE=1                        WGG00980
    IF (T0.LE.TLIM) GOTO 100         WGG00990
    ICONE=0                          WGG01000
C*** FOR HC, BREAK PATH UP INTO TWO PARTS. FIRST RISE PORTION CONE, WGG01010
    THEN POST-RISE FRUSTRUM.          WGG01020
    X0=XLIM                         WGG01030
    Y0=YLIM                         WGG01040
    Z0=ZLIM                         WGG01050
    GOTO 150                         WGG01060
220   IF (ICONE.EQ.1) RETURN        WGG01070
    ICONE=1                         WGG01080
C*** NOW POST RISE PORTION,        WGG01090
    IF (PATHL.EQ.0.) GOTO 100        WGG01100
    IF (IFLAG(1).NE.3.AND.IFLAG(2).NE.3) RETURN      WGG01110
    IF (IFLAG(1).EQ.3.AND.IFLAG(2).EQ.3) RETURN      WGG01120
    K=1                            WGG01130
    IF (IFLAG(2).EQ.3) K=2          WGG01140
    J=1                            WGG01150
    IF (XL<2>.LT.XL<1>) J=2        WGG01160
    XL<J>=XINT(K)                 WGG01170
    YL<J>=YINT(K)                 WGG01180
    ZL<J>=ZINT(K)                 WGG01190
    X0=A                           WGG01200
    Y0=B                           WGG01210
    Z0=C                           WGG01220
    CALL YYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)      WGG01230
    PATHL=PATHL+SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+    WGG01240
    *(ZINT(1)-ZINT(2))**2)
    RETURN                         WGG01250
    END                           WGG01260
                                         WGG01270
                                         WGG01280

```

```

FUNCTION QROOT(ISIGN,A,B,C)          QR0000010
C*****PURPOSE:                      QR0000020
      TO FIND ROOTS OF A QUADRATIC EQUATION.
      IF (A.EQ.0.) GOTO 2              QR0000030
      IF (A.EQ.0.) GOTO 2              QR0000040
      XX=1.0*ISIGN                    QR0000050
      TEST=B*B - 4.0*A*C             QR0000060
      IF (TEST.LT.0.0) GO TO 1        QR0000070
      QROOT=(-1.0*B + XX*SQRT(TEST))/(2.0*A)
      GO TO 100                      QR0000080
1     QROOT=0.0                      QR0000090
      GOTO 100                      QR0000100
2     IF (B.EQ.0.) GOTO 1          QR0000110
      QROOT=-C/B                      QR0000120
100   RETURN                         QR0000130
      END                           QR0000140
                                  QR0000150

```

```

SUBROUTINE XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG) XYZ200010
DIMENSION XL(2),YL(2),ZL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2) XYZ200020
DIMENSION DIST(2),TEST(2) XYZ200030
*****SUBROUTINE XYZINT*****
*****PURPOSE:
      TO FIND THE X,Y,Z INTERCEPTS OF A TARGET-OBSERVER LINE OF SIGHT XYZ200040
      WITH A SMOKE CLOUD DESCRIBED BY: XYZ200050
      NCODE = 1 A HALF ELLIPTIC CONE WITH APEX AT THE ORIGIN XYZ200070
              AND LEADING EDGE TRUNCATION BY THE X=X0 PLANE XYZ200080
              AND BOTTOM EDGE TRUNCATION BY THE Z=0 PLANE. XYZ200090
      NCODE = 2 A QUARTER ELLIPSOID WITH APEX XYZ200100
              AT THE ORIGIN AND WITH LEADING EDGE TRUNCATED BY XYZ200110
              THE X=X0 AND BOTTOM EDGE TRUNCATED BY THE Z=0 PLANE. XYZ200120
*****INPUT:
      XL(2),YL(2),ZL(2) = X,Y,Z COORDINATES OF TWO POINTS THROUGH XYZ200130
      WHICH THE LOS PASSES(IE. TARGET AND OBSERVER COORDINATES). XYZ200140
      X0,Y0,Z0 = LENGTH OF SEMI-AXES OF ELLIPSOID. XYZ200150
*****OUTPUT:
      XINT(2),YINT(2),ZINT(2) = X,Y,Z COORDINATES OF THE INTERCEPTS XYZ200160
      OF THE LOS WITH THE ELLIPSOID. IFLAG(2) = INTERCEPT TYPE FOR EACH INTERCEPT COORD: XYZ200170
      = 0 NO INTERCEPT XYZ200180
      = 1 INTERIOR TO VOLUME XYZ200190
      = 2 ON CONICAL OR ELLIPTICAL SURFACE XYZ200200
      = 3 ON LEADING EDGE OF SURFACE XYZ200210
*****MATHEMATICAL APPROACH:
      THE EQUATION OF THE ELLIPSOID CAN BE WRITTEN AS: XYZ200220
       $(X-X0)^2 + (Y-Y0)^2 + (Z-Z0)^2 = 1$  XYZ200230
      AND THE EQUATION OF THE LOS CAN BE WRITTEN AS: XYZ200240
       $(X-XL1)/(XL2-XL1) = (Y-YL1)/(YL2-YL1) = (Z-ZL1)/(ZL2-ZL1)$  XYZ200250
      THE TWO EQUATIONS ARE COMBINED TO FORM A QUADRATIC EQUATION XYZ200260
      WHICH IS SOLVED TO GIVE THE INTERCEPTS. XYZ200270
      SIMILARLY FOR THE LOS EQUATIONS AND ELLIPTIC CONE : XYZ200280
       $(Z-Z0)^2 + (Y-Y0)^2 - (X-X0)^2 = 0$  XYZ200290
*****SPECIAL NOTES
      (1) WHEN TWO OR MORE COORDINATES ARE THE SAME, SPECIAL CASES ARE XYZ200300
      FORMED WHICH MUST BE DEALT WITH SEPARATELY BECAUSE OF XYZ200310
      SINGULARITIES IN THE LOS EQUATION. XYZ200320
      (2) WHEN TARGET AND/OR OBSERVER ARE INSIDE THE CLOUD INTERCEPTS XYZ200330
      ARE TAKEN AS THE TARGET AND/OR OBSERVER COORDINATES. XYZ200340
      (3) PROPER ACCOUNT IS TAKEN FOR A LOS INTERCEPTING THE CLOUD XYZ200350
      LEADING EDGE BUT XYZ200360
      (4) ALL COORDINATES MUST BE ABOVE THE Z=0 PLANE (IE. ABOVE THE XYZ200370
      SURFACE.) XYZ200380
*****SUBROUTINE CALLED... QR0OT
*****INITIALIZE PROGRAM VARIABLES
      I1=0 XYZ200450
      I2=0 XYZ200460
      INT=0 XYZ200470
      LEAD=0 XYZ200480
      ISURF=0 XYZ200490
      K1=0 XYZ200500
      K2=0 XYZ200510
      K3=0 XYZ200520
      K4=0 XYZ200530
      TEST(1)=0.0 XYZ200540
      TEST(2)=0.0 XYZ200550
      TEST3=0.0 XYZ200560
      TEST4=0.0 XYZ200570
      DIST(1)=0.0 XYZ200580
      DIST(2)=0.0 XYZ200590
      DELX=XL(2)-XL(1) XYZ200600
      DELY=YL(2)-YL(1) XYZ200610
      DELZ=ZL(2)-ZL(1) XYZ200620
      IFLAG(1)=0 XYZ200630
      IFLAG(2)=0 XYZ200640
      XYZ200650
      XYZ200660
      XYZ200670
      XYZ200680
      XYZ200690
      XYZ200700
C*** REJECT IMMEDIATELY IF BOTH TGT/OBS BELOW Z=0.
      IF (ZL(1).LT.0. AND. ZL(2).LT.0.) GOTO 800
*****DETERMINE SPECIAL CASES FOR LOS

```

```

ICASE=1          XYZ000710
IF <ABS(DELX)>.GT.<.01*ABS(DELAY)>,AND.   XYZ000720
*    ABS(DELX).GT.ABS(.01*ABS(DELZ))) GOTO 12  XYZ000730
ICASE=2          XYZ000740
IF <ABS(DELAY)>.GT.<.01*ABS(DELZ))> GOTO 12  XYZ000750
ICASE=3          XYZ000760
IF <DELZ.GT.1.E-2.OR.DELZ.LT.-1.E-2> GOTO 12  XYZ000770
ICASE=4          XYZ000780
12 CONTINUE      XYZ000790
C*****DEFAULT SPECIAL CASE OF OBS-TAR COINCIDENT  XYZ000800
IF(ICASE.NE.4)GO TO 14  XYZ000810
GO TO 800        XYZ000820
14 CONTINUE      XYZ000830
C*****SET UP TEST TO DETERMINE IF TARGET AND/OR OBSERVER ARE IN THE  XYZ000840
C INTERIOR OF THE CLOUD  XYZ000850
DO 1 I=1,2      XYZ000860
IF <NCODE.EQ.2> TEST(I)= <(XL(I)-X0)/X0>**2 + <(YL(I)/Y0)>**2 +  XYZ000870
*<(ZL(I)/Z0)>**2-1.  XYZ000880
IF <NCODE.EQ.1> TEST(I)=<(YL(I)/Y0)>**2 + <(ZL(I)/Z0)>**2 -  XYZ000890
*<(XL(I)/X0)>**2  XYZ000900
1 CONTINUE      XYZ000910
IF<TEST(1).GT.0.0>GO TO 2  XYZ000920
IF <XL(1).LT.0.,OR,ZL(1).LT.0. > GOTO 2  XYZ000930
IF <XL(1).GT.X0> GOTO 2  XYZ000940
IF <XL(1).EQ.X0> K1=1  XYZ000950
IF <TEST(1).EQ.0.,OR,ZL(1).EQ.0. > K3=1  XYZ000960
I1=1            XYZ000970
2 CONTINUE      XYZ000980
IF<TEST(2).GT.0.0>GO TO 3  XYZ000990
IF <XL(2).LT.0.,OR,ZL(2).LT.0. > GOTO 3  XYZ01000
IF <XL(2).GT.X0> GOTO 3  XYZ01010
IF <XL(2).EQ.X0> K2=1  XYZ01020
IF <TEST(2).EQ.0..OR.ZL(2).EQ.0. > K4=1  XYZ01030
I2=1            XYZ01040
C*****IF BOTH POINTS ARE IN THE CLOUD SET INTERCEPTS EQUAL TO THE  XYZ01050
C TARGET-OBSERVER COORDINATES AND RETURN--OTHERWISE CONTINUE  XYZ01060
C
3 IF<I1.EQ.0>GO TO 4  XYZ01070
IF<I2.EQ.0>GO TO 4  XYZ01080
DO 5 I=1,2      XYZ01090
IFLAG(I)=1      XYZ01100
XINT(I)=XL(I)  XYZ01110
YINT(I)=YL(I)  XYZ01120
5 ZINT(I)=ZL(I)  XYZ01130
IF <K3.EQ.1> IFLAG(1)=2  XYZ01140
IF <K4.EQ.1> IFLAG(2)=2  XYZ01150
IF <K1.EQ.1> IFLAG(1)=3  XYZ01160
IF <K2.EQ.1> IFLAG(2)=3  XYZ01170
GO TO 999       XYZ01180
C*****IF ONLY ONE POINT IS IN CLOUD KEEP TRACK OF IT FOR LATER  XYZ01190
4 CONTINUE      XYZ01200
IF<I1.EQ.0>GO TO 6  XYZ01210
INT=1           XYZ01220
6 IF<I2.EQ.0>GO TO 7  XYZ01230
INT=2           XYZ01240
7 CONTINUE      XYZ01250
IF <K1.EQ.1> LEAD=1  XYZ01260
IF <K2.EQ.1> LEAD=2  XYZ01270
IF <K3.EQ.1> ISURF=1  XYZ01280
IF <K4.EQ.1> ISURF=2  XYZ01290
C*****SET UP LOS EQUATION DEPENDING UPON CASE  XYZ01300
GO TO <10,20,30>,ICASE  XYZ01310
C*****CASE 1      XYZ01320
10 SX=DELX/DELX  XYZ01330
SY=DELY/DELX  XYZ01340
SZ=DELZ/DELX  XYZ01350
XI=XL(1)-SX*XL(1)  XYZ01360
YI=YL(1)-SY*XL(1)  XYZ01370
ZI=ZL(1)-SZ*XL(1)  XYZ01380
GO TO 101       XYZ01390
C*****CASE 2      XYZ01400

```

```

20 SX=DELX/DELY
SY=DELY/DELY
SZ=DELZ/DELY
XI=XL(1)-SX*YL(1)
YI=YL(1)-SY*YL(1)
ZI=ZL(1)-SZ*YL(1)
GO TO 101
C*****CASE 3
30 SX=DELX/DELZ
SY=DELY/DELZ
SZ=DELZ/DELZ
XI=XL(1)-SX*ZL(1)
YI=YL(1)-SY*ZL(1)
ZI=ZL(1)-SZ*ZL(1)
101 CONTINUE
C*****SET UP QUADRATIC COEFFICIENTS
IF (NCODE.EQ.1) GOTO 60
A=(SX/X0)**2 + (SY/Y0)**2 + (SZ/Z0)**2
B=2.0*(XI/X0)*(SX/X0) + (YI/Y0)*(SY/Y0) + (ZI/Z0)*(SZ/Z0)
*- (SX/X0)
C=(XI/X0)**2 + (YI/Y0)**2 + (ZI/Z0)**2 - 2.0*XI/X0
GOTO 61
60 A=(SY/Y0)**2 + (SZ/Z0)**2 - (SX/X0)**2
B=2.0*(SY/Y0)*(YI/Y0) + (SZ/Z0)*(ZI/Z0) + (SX/X0)*(XI/X0)
C=(YI/Y0)**2 + (ZI/Z0)**2 - (XI/X0)**2
61 CONTINUE
C*****DEFAULT ALL INTERCEPTS IF ROOTS ARE COMPLEX
TEST0=B*B-4.0*A*C
IF (TEST0.GE.0.0) GO TO 888
800 DO 13 I=1,2
IFLAG(I)=0
XINT(I)=0.0
YINT(I)=0.0
13 ZINT(I)=0.0
GO TO 999
C*****SOLVE QUADRATIC FOR X,Y OR Z DEPENDING ON CASE
888 GO TO (100,200,300),ICASE
100 XINT(1)=QRROOT(+1,A,B,C)
XINT(2)=QRROOT(-1,A,B,C)
DO 11 I=1,2
YINT(I)=YI+SY*XINT(I)
11 ZINT(I)=ZI+SZ*XINT(I)
GOTO 400
200 YINT(1)=QRROOT(+1,A,B,C)
YINT(2)=QRROOT(-1,A,B,C)
DO 21 I=1,2
XINT(I)=XI+SX*YINT(I)
21 ZINT(I)=ZI+SZ*YINT(I)
GO TO 400
300 ZINT(1)=QRROOT(+1,A,B,C)
ZINT(2)=QRROOT(-1,A,B,C)
DO 31 I=1,2
XINT(I)=XI+SX*ZINT(I)
31 YINT(I)=YI+SY*ZINT(I)
C*** TEST FOR VALID INTERCEPTS
400 I1=0
I2=0
IFLAG(1)=2
IFLAG(2)=2
IF (ZINT(1).GE.0..AND.XINT(1).GE.0..AND.XINT(1).LE.X0) I1=1
IF (ZINT(2).GE.0..AND.XINT(2).GE.0..AND.XINT(2).LE.X0) I2=1
IF (I1.EQ.0.OR.I2.EQ.0) GOTO 450
IF (XINT(1).EQ.XINT(2).AND.YINT(1).EQ.YINT(2).AND.ZINT(1).EQ.
* ZINT(2)) I2=0
IF (I1.EQ.1.AND.I2.EQ.1) GOTO 600
C*** AT LEAST ONE INTERCEPT INVALID. FIRST COMPUTE POSSIBLE Z=0, INTcpt
450 GOTO (460,470,480),ICASE
460 Z=0
IF (SZ.EQ.0.) GOTO 500
X=-ZI/SZ

```

```

Y=YI+SY*X
GOTO 490
470 Z=0
IF <SZ,EQ.0.,> GOTO 500
Y=-ZI/SZ
X=XI+SX*Y
GOTO 490
480 Z=0
X=XI
Y=YI
C*** TEST IF Z=0, INTERCEPT VALID,
490 IF <X,GT,X0,OR,X,LT,0.,> GOTO 500
IF <NCODE,EQ,1,> TEST3=(Y/Y0)**2-(X/X0)**2
IF <NCODE,EQ,2,> TEST3=<(X-X0)/X0>**2 + <(Y/Y0)>**2 - 1.
IF <TEST3,GT,0.,> GOTO 500
IF <I1,EQ,1,> GOTO 495
XINT(1)=X
YINT(1)=Y
ZINT(1)=Z
I1=1
IF <I2,EQ,1,> GOTO 600
GOTO 500
495 XINT(2)=X
YINT(2)=Y
ZINT(2)=Z
I2=1
IF <I1,EQ,1,> GOTO 600
C*** NEXT COMPUTE INTERCEPT ON X=X0
500 GOTO (510,505,507),ICASE
505 X=X0
Y=(X-XI)/SX
Z=ZI+SZ*Y
GOTO 515
507 X=X0
Z=(X-XI)/SX
Y=YI+SY*Z
GOTO 515
510 X=X0
Y=YI+SY*X
Z=ZI+SZ*X
515 IF <Z,LT,0.,> GOTO 800
TEST4=(Y/Y0)**2 + <Z/Z0>**2 - 1.
IF <TEST4,GT,0.,> GOTO 800
IF <I1,EG,1,> GOTO 520
XINT(1)=X
YINT(1)=Y
ZINT(1)=Z
I1=1
IF <I2,EQ,0,> GOTO 800
GOTO 600
520 XINT(2)=X
YINT(2)=Y
ZINT(2)=Z
I2=1
600 DO 620 I=1,2
X1=<(XL(1)-XINT(I))**2 + (YL(1)-YINT(I))**2 + (ZL(1)-ZINT(I))**2
X2=<(XL(2)-XINT(I))**2 + (YL(2)-YINT(I))**2 + (ZL(2)-ZINT(I))**2
DIST(I)=SQRT(X1+SQRT(X2))
620 CONTINUE
TEST5=1.0001*SQRT(DELX**2+DELY**2+DELZ**2)
IF(DIST(1).GT.TEST5,AND,DIST(2).GT.TEST5) GOTO 800
IF<INT.EQ,0,> GOTO ?00
C*** REPLACE ONE INTERCEPT WITH INTERIOR OBS/TGT.
DO 610 I=1,2
IF <XL(INT).GT.<(XINT(I)+.001).OR.XL(INT).LT.<(XINT(I)-.001)>
* GOTO 610
IF <YL(INT).GT.<(YINT(I)+.001).OR.YL(INT).LT.<(YINT(I)-.001)>
* GOTO 610
IF <ZL(INT).GT.<(ZINT(I)+.001).OR.ZL(INT).LT.<(ZINT(I)-.001)>
* GOTO 610
XYZ02110
XYZ02120
XYZ02130
XYZ02140
XYZ02150
XYZ02160
XYZ02170
XYZ02180
XYZ02190
XYZ02200
XYZ02210
XYZ02220
XYZ02230
XYZ02240
XYZ02250
XYZ02260
XYZ02270
XYZ02280
XYZ02290
XYZ02300
XYZ02310
XYZ02320
XYZ02330
XYZ02340
XYZ02350
XYZ02360
XYZ02370
XYZ02380
XYZ02390
XYZ02400
XYZ02410
XYZ02420
XYZ02430
XYZ02440
XYZ02450
XYZ02460
XYZ02470
XYZ02480
XYZ02490
XYZ02500
XYZ02510
XYZ02520
XYZ02530
XYZ02540
XYZ02550
XYZ02560
XYZ02570
XYZ02580
XYZ02590
XYZ02600
XYZ02610
XYZ02620
XYZ02630
XYZ02640
XYZ02650
XYZ02660
XYZ02670
XYZ02680
XYZ02690
XYZ02700
XYZ02710
XYZ02720
XYZ02730
XYZ02740
XYZ02750
XYZ02760
XYZ02770
XYZ02780
XYZ02790
XYZ02800

```

```

IFLAG(I)=1
IF (<LEAD,EQ,INT>) IFLAG(I)=3
IF (<ISURF,EQ,INT>) IFLAG(I)=2
GOTO 700
CONTINUE
C*** REPLACE CLOSEST INTERCEPT OUTSIDE LOS RANGE
K1=1
IF (<DIST(1),LT,DIST(2)>) K1=2
XINT(K1)=XL(INT)
YINT(K1)=YL(INT)
ZINT(K1)=ZL(INT)
IFLAG(K1)=1
IF (<XINT(1),LE,(X0-.001),AND,XINT(1),LE,(X0+.001)>) IFLAG(1)=3
IF (<XINT(2),LE,(X0-.001),AND,XINT(2),LE,(X0+.001)>) IFLAG(2)=3
999 RETURN
END

```

```

SUBROUTINE BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,
*BRA1,BRAT2,BRAT3,BRAT4,BRAT5)          BRA00010
                                         BRA00020
                                         BRA00030
                                         BRA00040
                                         BRA00050
                                         BRA00060
                                         BRA00070
                                         BRA00080
                                         BRA00090
                                         BRA00100
                                         BRA00110
                                         BRA00120
                                         BRA00130
                                         BRA00140
                                         BRA00150
                                         BRA00160
                                         BRA00170
                                         BRA00180
                                         BRA00190
                                         BRA00200
                                         BRA00210
                                         BRA00220
                                         BRA00230
                                         BRA00240
                                         BRA00250
                                         BRA00260
                                         BRA00270
                                         BRA00280
                                         BRA00290
                                         BRA00300
                                         BRA00310
                                         BRA00320
                                         BRA00330
                                         BRA00340
                                         BRA00350
                                         BRA00360
                                         BRA00370
                                         BRA00380
                                         BRA00390
                                         BRA00400
                                         BRA00410
                                         BRA00420
                                         BRA00430
                                         BRA00440
                                         BRA00450
                                         BRA00460
                                         BRA00470
                                         BRA00480
                                         BRA00490
                                         BRA00500
                                         BRA00510
                                         BRA00520

THIS ROUTINE PROVIDES DEFAULT MUNITION CHARACTERISTIC VALUES TO
SMOKE. SMOKE MUNITION TYPES (TYPM) ARE GIVEN IN THE COMMENTS
OF THE MAIN ROUTINE IN SMOKE

DIMENSION B1(21),B2(21),B3(21),B4(21),B5(21)          BRA00010
DIMENSION F(21),T(21),E(21),IT(21)                  BRA00020
DATA B1 / .537, .631, .2218, .537, .2218, 1, 1, 1, 1, 1, 1, 1, 1, 0,          BRA00030
*   0, .521, 1, 631, 1, 808, .1204, .653, 1, 731, 0, 1, 1, 1, 1, 1, 1, 0,          BRA00040
*   0, .476, -.4985, 3, 915, .476, 3, 915, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00050
*   2, 106, .678, -2, 556, 3, 1012, -3, 136, -2, 852, 3, 6832, 0, 0, 0, 0, 0,          BRA00060
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00070
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00080
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00090
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00100
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00110
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00120
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00130
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00140
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00150
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00160
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00170
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00180
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00190
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00200
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00210
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00220
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00230
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00240
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00250
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00260
*   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,          BRA00270
MAXS=21
ITP=IFIX(TYPM+.0001)
IF (< ITP.EQ.0) RETURN
IF (< ITP.LE.MAXS) GOTO 10
IERR=1
RETURN
10  BRAT1=B1(ITP)
    BRAT2=B2(ITP)
    BRAT3=B3(ITP)
    BRAT4=B4(ITP)
    BRAT5=B5(ITP)
    ITYPE=ITC(ITP)
    IF (< MUNRD.NE.0) GOTO 20
    XN=1.
    YF=0.
    TBURN=T(ITP)
    EFF=E(ITP)
    FW=F(ITP)
    RETURN
20  IF (< XN.EQ.0.) XN=1.
    IF (< TBURN.EQ.0.) TBURN=T(ITP)
    IF (< EFF.EQ.0.) EFF=E(ITP)
    IF (< FW.EQ.0.) FW=F(ITP)
    RETURN
END

```

```

SUBROUTINE GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1,      GPU00010
*ZPP2)                                     GPU00020
GPU00030
THIS ROUTINE COMPUTES THE CL CONTRIBUTION FROM AN EXOTHERMIC,      GPU00040
BUOANTLY RISING SMOKE CLOUD OF GAUSSIAN DISTRIBUTION AND      GPU00050
UNIT CONCENTRATION. IT IS CENTERED ON THE LEADING EDGE AT      GPU00060
COORDINATES (A,B,C-2*SIGZ) WHERE      GPU00070
INPUTS   A = CLOUD DOWNWIND DISTANCE      GPU00080
        B = CLOUD BASE HALF-WIDTH (AT LEADING EDGE)      GPU00090
        C = CLOUD HEIGHT (AT LEADING EDGE)      GPU00100
        C2 = MOMENTUM RISE COEFFICIENT. (USUALLY BRIGGS OR LIMITED      GPU00110
              RISE BRIGGS)      GPU00120
XPP1,...ZPP2 COORDINATES OF TARGET AND OBSERVER IN MUNITION      GPU00130
CENRTED COORDINATE SYSTEM WITH WIND VECTOR X-AXIS      GPU00140
OUTPUT   CLGAUS = CL VALUE FOR UNIT MASS (METER**-2)      GPU00150
KCALL = SET TO 1 AFTER EVERY CHANGE IN TGT/OBS COORD.      GPU00160
DIMENSION AV(2),AP(2),BP(2),CP(2)      GPU00170
CLGAUS=0,      GPU00180
IF (KCALL.NE.0) GOTO 100      GPU00190
KCALL=1      GPU00200
ICASE=0      GPU00210
C*** COMPUTE LOS GENERALIZED COORDINATES.      GPU00220
IF (ZPP1.LT.0. .AND. ZPP2.LT.0.) RETURN      GPU00230
DELX=XPP2-XPP1      GPU00240
DELY=YPP2-YPP1      GPU00250
DELZ=ZPP2-ZPP1      GPU00260
IF (ABS(DELX).LE..01*ABS(DELY) .OR. ABS(DELX).LE..01*ABS(DELZ))      GPU00270
  , GOTO 10      GPU00280
  ICASE=1      GPU00290
  SX=DELX/DELX      GPU00300
  SY=DELY/DELX      GPU00310
  SZ=DELZ/DELX      GPU00320
  XI=XPP1-SX*XPP1      GPU00330
  YI=YPP1-SY*YPP1      GPU00340
  ZI=ZPP1-SZ*ZPP1      GPU00350
  AV(1)=XPP1      GPU00360
  AV(2)=XPP2      GPU00370
  GOTO 80      GPU00380
10 IF (ABS(DELY).LE..01*ABS(DELZ)) GOTO 20      GPU00390
  ICASE=2      GPU00400
  SX=DELX/DELY      GPU00410
  SY=DELY/DELY      GPU00420
  SZ=DELZ/DELY      GPU00430
  XI=XPP1-SX*YPP1      GPU00440
  YI=YPP1-SY*YPP1      GPU00450
  ZI=ZPP1-SZ*ZPP1      GPU00460
  AV(1)=YPP1      GPU00470
  AV(2)=YPP2      GPU00480
  GOTO 80      GPU00490
20 IF (ABS(DELZ).LT..001) RETURN      GPU00500
  ICASE=3      GPU00510
  SX=DELX/DELZ      GPU00520
  SY=DELY/DELZ      GPU00530
  SZ=DELZ/DELZ      GPU00540
  XI=XPP1-SX*ZPP1      GPU00550
  YI=YPP1-SY*ZPP1      GPU00560
  ZI=ZPP1-SZ*ZPP1      GPU00570
  AV(1)=ZPP1      GPU00580
  AV(2)=ZPP2      GPU00590
80 IF (ZPP1.LT.0.) AV(1)=-ZI/SZ      GPU00600
  IF (ZPP2.LT.0.) AV(2)=-ZI/SZ      GPU00610
  SMUL=SQRT(SX*SX+SY*SY+SZ*SZ)      GPU00620
100 IF (ICASE.EQ.0) RETURN      GPU00630
C*** COMPUTE GAUSSIAN PARAMETERS, REAL AND REFLECTED IMAGE CLOUD TO      GPU00640
C ACCOUNT FOR GROUND REFLECTED SMOKE      GPU00650
  SIGZ=(2.73+C2*A)/2.15      GPU00660
  ZB=C-(2.73+C2*A)      GPU00670
  IF (ZB.GE.C) RETURN      GPU00680
  SIGY=B*SQRT(1.-((ZB/C)**2)/2.15)      GPU00690
  IF (ZB.LT.0.) SIGY=B/2.15      GPU00700

```

```

SIGX=SIGY
ASIC=(SX/SIGX)**2 + (SY/SIGY)**2 + (SZ/SIGZ)**2
BMEAN=2.*((SX*(XI-A)/SIGX**2) + (SY*YI/SIGY**2))
BP(1)=BMEAN+2.*SZ*(ZI-ZB)/SIGZ**2
GPU00710 GPU00720 GPU00730 GPU00740 GPU00750 GPU00760 GPU00770 GPU00780 GPU00790 GPU00800 GPU00810 GPU00820 GPU00830 GPU00840 GPU00850 GPU00860 GPU00870 GPU00880 GPU00890 GPU00900 GPU00910 GPU00920 GPU00930 GPU00940 GPU00950 GPU00960 GPU00970 GPU00980 GPU00990 GPU01000 GPU01010
BP(2)=BMEAN+2.*SZ*(ZI+ZB)/SIGZ**2
CTOT=((XI-A)/SIGX)**2 + (YI/SIGY)**2
CP(1)=CTOT + ((ZI-ZB)/SIGZ)**2
CP(2)=CTOT + ((ZI+ZB)/SIGZ)**2
C*** CALCULATE FOR LOS INTEGRAL
CMUL=SMUL/(2.*3.14159*SIGX*SIGY*SIGZ*SQRT(ASIC))
DO 220 I=1,2
CEXU=.5*(CP(I)-BP(I)**2)/(4.*ASIC)
IF (CEXU.GT.20.) GOTO 220
C*** INFINITE PATH LOS
CLU=EXP(-CEXU)
C*** CORRECTION FOR FINITE PATH
DO 210 J=1,2
AP1=(AV(J)+BP(J)/(2.*ASIC))*SQRT(ASIC/2.)
P1=ABS(AP1)
CP1=0
IF (P1.LE.5.) CP1=0.5/(1.+P1*(.0705230784+P1*(.042282013+P1*(.0092705272+P1*(.0001520134+P1*(.0002765672+P1*.0000430638))))))>0
***16
IF (AP1.GE.0.) CP1=1.-CP1
AP(J)=CP1
210 CONTINUE
CLGAUS=CLGAUS+CLU*ABS(AP(2)-AP(1))
220 CONTINUE
CLGAUS=CMUL*CLGAUS
RETURN
END

```

SUBROUTINE LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)

C THIS SUBROUTINE CALCULATES MOLECULAR ABSORPTION COEFFICIENTS AT
 LASER FREQUENCIES. PH20 AND T ARE THE PARTIAL PRESSURE OF WATER
 VAPOR AND TEMPERATURE, IN TORR AND DEGREES K RESPECTIVELY.
 LID IS THE LASER LINE IDENTIFICATION AS DESCRIBED BELOW.
 ABCDEF IS THE ABSORPTION COEFFICIENT RETURNED IN KM-1.

 INPUT
 CARD 1 LNAME1, LNAME2, PH20, T, LZPATH FORMAT (2A4,3F10.3)
 LNAME1 FIRST 4 CHARS OF LASER LINE (A4)
 LNAME2 SECOND 4 CHARS OF LASER LINE (A4)
 ***IF LNAME NOT ENTERED WILL USE WAVELENGTH READ IN MAIN
 PH20 WATER VAPOR PRESSURE 0. TO .35 (MB) (F10.3)
 T AMBIENT AIR TEMPERATURE 260. TO .320 (C) (F10.3)
 LZPATH PATHLENGTH IN KM F(10.3)
 ***PH20, AND T NOT REQUIRED WHEN ICLIMATE(IN MAIN)=1 OR 2

 OUTPUT

LZTRAN TRANSMISSION

NOTES
 ABCDEF RETURNS THE ABSORPTION COEFFICIENT (KM-1)

LNAME1 FIRST 4 CHARS OF LASER LINE ON NORMAL RETURN
 BLANK ON ERROR RETURN

LNAME2 SECOND 4 CHARS OF LASER LINE ON NORMAL RETURN
 BLANK ON ERROR RETURN

++ CALLED PROGRAMS ++

LZDNM

 LASER LINE IDENTIFICATION

LID=1 ND:YAG LASER, 1.06 MICRONS
 LID=2 CO2 LASER LINE P(20), 10.591 MICRONS
 LID=101 TO 127 DF LASER, 3.521 TO 4.089 MICRONS
 101 P3(12) * 107 P3(8) * 113 P3(5) * 119 P2(5) * 125 P1(5)
 102 P3(11) * 108 P2(11) * 114 P2(8) * 120 P1(8) * 126 P1(4)
 103 P3(10) * 109 P3(?) * 115 P2(?) * 121 P2(4) * 127 P1(3)
 104 P2(13) * 110 P2(10) * 116 P1(10) * 122 P1(7) *
 105 P3(9) * 111 P3(6) * 117 P2(6) * 123 P2(3) *
 106 P2(12) * 112 P2(9) * 118 P1(9) * 124 P1(6) *
 LID=201 TO 219 CO LASER, 4.908 TO 5.088 MICRONS
 201 P6(12) * 205 P6(8) * 209 P5(12) * 213 P5(8) * 217 P4(9)
 202 P6(11) * 206 P5(15) * 210 P5(11) * 214 P5(7) * 218 P4(8)
 203 P6(10) * 207 P5(14) * 211 P5(10) * 215 P4(11) * 219 P4(7)
 204 P6(9) * 208 P5(13) * 212 P5(9) * 216 P4(10) *
 LID=301 TO 305 GA AS LASER, (GA.85 TO GA.950) LASER LINE NAMES
 301 0.850 MICROMETERS * 304 0.925 MICROMETERS
 302 0.875 MICROMETERS * 305 0.950 MICROMETERS
 303 0.900 MICROMETERS *
 INTEGER LNAME1,LNAME2,BLANK,LNAME3,LNAME4
 REAL LAZTRN,LZPATH
 DIMENSION ADF0(30),ADF1(30),ADF2(30),ADF3(30),ADF4(30)
 ,ADF5(30)
 DIMENSION AC00(20),AC01(20),AC02(20),AC03(20),AC04(20)
 ,AC05(20)
 DIMENSION AGA0(5),AGA1(5),AGA2(5),AGA3(5),AGA4(5),
 ,AGA5(5),AGA6(5),AGA7(5),AGA8(5)
 COMMON /CONST/PI,PI2,PIRAD,TWOP,TORRMB,CDEGK
 COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
 ,FOGPRB,WNDVEL,WNDDIR,IPASCT

LZT00010
 LZT00020
 LZT00030
 LZT00040
 LZT00050
 LZT00060
 LZT00070
 LZT00080
 LZT00090
 LZT00100
 LZT00110
 LZT00120
 LZT00130
 LZT00140
 LZT00150
 LZT00160
 LZT00170
 LZT00180
 LZT00190
 LZT00200
 LZT00210
 LZT00220
 LZT00230
 LZT00240
 LZT00250
 LZT00260
 LZT00270
 LZT00280
 LZT00290
 LZT00300
 LZT00310
 LZT00320
 LZT00330
 LZT00340
 LZT00350
 LZT00360
 LZT00370
 LZT00380
 LZT00390
 LZT00400
 LZT00410
 LZT00420
 LZT00430
 LZT00440
 LZT00450
 LZT00460
 LZT00470
 LZT00480
 LZT00490
 LZT00500
 LZT00510
 LZT00520
 LZT00530
 LZT00540
 LZT00550
 LZT00560
 LZT00570
 LZT00580
 LZT00590
 LZT00600
 LZT00610
 LZT00620
 LZT00630
 LZT00640
 LZT00650
 LZT00660
 LZT00670
 LZT00680
 LZT00690
 LZT00700

COMMON /IOUNIT/I0UNIT, I0IN, I0OUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTULZT00710
 COMMON /GEOMET/PTS(15), IGEOISW
 C THE POLYNOMIAL COEFFICIENTS ARE SELECTED BY THE LID. THE LZT00720
 C INDEX FOR THE COEFFICIENT ARRAYS FOR THE DF LASER IS LZT00740
 C I = LID - 100. NOW I IS IN THE RANGE 1..27 SINCE THERE ARE LZT00760
 C 27 DF LASER LINES. WHEN THE POLYNOMIAL IS EVALUATED I IS LZT00780
 C USED TO INDEX THE ARRAYS ADF/0..5/ THUS SELECTING THE LZT00790
 C CORRECT COEFFICIENTS FOR THE LASER LINE SELECTED. LZT00800
 C COEFFICIENTS FOR THE OTHER LASER POLYNOMIALS ARE SELECTED LZT00810
 C IN THE SAME FASHION.
 C POLYNOMIAL COEFFICIENTS FOR DF LASER LINES, (1..27) LZT00820
 DATA ADF0/.1019, .08352, .04083, .03675, .02042, .01833, LZT00830
 1 .04738, .03134, .07870, .05844, .1096E-2, .9353E- LZT00840
 2 , .2537E-2, .3254E-2, -.1103E-2, -.6471E-3, -.1423E- LZT00850
 3 , -.4664E-2, -.1221E-4, -.1698E-3, .1172E-3, .6195E- LZT00860
 4 , .1272E-2, .5485E-2, .1651E-2, .6913E-2, -.4498E- LZT00870
 5 , 2*3*0./
 DATA ADF1/- .9718E-4, -.1160E-3, -.4892E-4, -.4230E-4, -. LZT00880
 1 .1750E-4, -.1524E-4, -.5589E-4, -.4642E-4, -.1218E- LZT00890
 2 , .3683E-4, .1672E-4, -.1346E-4, -.4765E-6, -.8548E- LZT00900
 3 , .6089E-5, .8897E-5, .1507E-5, .4448E-4, .4540E- LZT00910
 4 , .5569E-6, .1327E-6, .1462E-4, .7524E-6, -.1398E- LZT00920
 5 , .7023E-6, -.1044E-4, .1816E-4, 3*0./ LZT00930
 DATA ADF2/- .9666E-2, .7252E-2, .7050E-2, .7142E-2, .6320E- LZT00940
 1 , .6191E-2, .5400E-2, .5344E-2, .4064E-2, .4682E- LZT00950
 2 , .3734E-2, .5839E-2, .5075E-2, .2484E-2, .8190E- LZT00960
 3 , .6920E-2, .8779E-2, .7914E-2, .7094E-2, .5327E- LZT00970
 4 , .6692E-2, .9452E-2, .01025, .01367, .01279, .7844E- LZT00980
 5 , .01201, 3*0./
 DATA ADF3/- .2655E-4, -.1805E-4, -.1879E-4, -.1937E-4, -. LZT01000
 1 .1704E-4, -.1669E-4, -.1412E-4, -.1417E-4, -.9076E- LZT01010
 2 , .1165E-4, .7371E-5, .9113E-5, .1290E-4, -. LZT01020
 3 , .1070E-5, .6746E-5, .1573E-4, .1683E-4, .1883E- LZT01030
 4 , -.1774E-4, .3002E-4, -.1025E-4, .2489E-4, .2596E- LZT01040
 5 , .2848E-4, .2858E-4, .4991E-5, .2608E-4, 3*0./ LZT01050
 DATA ADF4/- .7847E-4, .5729E-4, .5585E-4, .5606E-4, .4847E- LZT01060
 1 , .4668E-4, .4145E-4, .4218E-4, .3314E-4, .3642E- LZT01070
 2 , .3170E-4, .3682E-4, .3835E-4, .4798E-4, .3537E- LZT01080
 3 , .4635E-4, .4941E-4, .5601E-4, .5494E-4, -.1120E- LZT01090
 4 , .6567E-4, .9104E-4, .7398E-4, .8509E-4, .8746E- LZT01100
 5 , .1050E-3, .1060E-3, 3*0./
 DATA ADF5/- .2056E-6, -.1374E-6, -.1408E-6, -.1432E-6, -. LZT01120
 1 .1222E-6, -.1172E-6, -.1011E-6, -.1054E-6, -.7169E- LZT01130
 2 , .8600E-7, .6588E-7, .8998E-7, .9081E-7, -. LZT01140
 3 , .1320E-6, .1570E-7, .9751E-7, .8159E-7, .1234E- LZT01150
 4 , .1313E-6, .1836E-6, .1469E-6, .2540E-6, .1796E- LZT01160
 5 , .1691E-6, .1886E-6, .2635E-6, .2855E-6, 3*0./ LZT01170
 C POLYNOMIAL COEFFICIENTS FOR CO LASER LINES, (1..19)
 DATA AC00/- .1.813E-3, -.9.289E-4, .1.153E-3, -.1.985E-3, -. LZT01180
 1 .4.523E-3, -.1.205E-3, -.2.225E-4, -.4.061E-3, -.4.522E- LZT01190
 2 , .2.267E-6, .5.917E-3, -.1.423E-3, .3.640E-3, .1.096E- LZT01200
 3 , .3.6.455E-4, -.3.9.222E-3, -.1.873E-5, -.1.055E-4, .1.489E- LZT01210
 4 , .2.0./
 DATA AC01/3 .426E-5, 3.658E-6, -.1.372E-6, 7.229E-6, 1.641E- LZT01220
 1 .5.6.423E-6, 1.042E-7, 1.435E-5, 1.593E-4, -.5.334E- LZT01230
 2 , 7.1.498E-5, 5.5.284E-6, 1.806E-5, -.3.651E-6, -.1.303E- LZT01240
 3 , 6.1.835E-5, 4.7555E-6, 2.330E-6, 6.196E-6, 0./
 DATA AC02/8 .813E-2, -.1.020E-1, .4.881E-2, .6.872E-2, -. LZT01250
 1 .6.244E-1, .4.474E-2, 1.226E-2, -.1.462E-2, 1.490E- LZT01260
 2 , 1.1.428E-2, 1.934E-9, .034E-3, -.1.091E-1, 1.284E-2, -. LZT01270
 3 , 2.1.31E-2, 6.543E-3, 2.824E-2, 9.463E-3, -.9.885E-2, 0./ LZT01280
 DATA AC03/8 .384E-4, 1.211E-3, -.4.687E-6, -.7.765E- LZT01290
 1 .5.2.641E-3, -.3.135E-5, 1.620E-4, 6.707E-4, 2.581E- LZT01300
 2 , 3.4.018E-4, -.3.113E-3, 9.692E-5, 7.907E-4, 1.070E- LZT01310
 3 , 4.3.026E-4, 1.216E-4, -.1.859E-5, 4.274E-5, 6.241E- LZT01320
 4 , 4.0./
 DATA AC04/2 .850E-4, -.4.934E-5, -.2.176E-5, 1.253E- LZT01330
 1 .4.1.158E-3, 1.926E-5, 9.823E-5, -.2.395E-4, -.6.255E- LZT01340
 2 , 5.6.630E-5, 4.851E-3, -.5.183E-5, -.6.993E-4, -.8.196E- LZT01350
 3 , 5.-2.239E-4, -.1.120E-4, 5.415E-5, -.8.393E-5, -.5.893E- LZT01360

```

4      4.0./          L2T01410
      DATA AC05/4.209E-6,3.426E-6,5.383E-7,6.252E-8,-4.149E-
1      6.2.374E-7,-8.637E-9,3.454E-6,1.526E-5,2.140E-6,- L2T01420
2      4.282E-6,5.433E-7,4.231E-6,7.778E-7,1.695E- L2T01430
3      6.8.830E-7,-1.548E-8,4.629E-7,3.241E-6,0./ L2T01440
C  POLYNOMIAL COEFFICIENTS FOR CO2 LASER           L2T01450
      DATA AC020,AC021,AC022,AC023,AC024,AC025,AC026,AC027, L2T01460
1      AC028/-4.4488,-4.1864E-3,5.7903E-2,-3.6770E- L2T01470
2      4.3.8521E-3,-4.7330E-6,1.0000E-5,6.0131E-7,- L2T01480
3      1.7441E-8/ L2T01490
C  POLYNOMIAL COEFFICIENTS FOR GA AS LASER, (1..5) L2T01500
      DATA AGA0/7.947E-3,7.590E-3,1.010,0.6094,4.271/ L2T01510
      DATA AGA1/-3.543E-5,-3.544E-5,-8.012E-3,-4.777E-3,- L2T01520
1      3.371E-2/ L2T01530
      DATA AGA2/1.740E-4,1.203E-4,8.736E-2,4.527E-2,.3364/ L2T01540
      DATA AGA3/-5.855E-7,-5.093E-7,-2.135E-4,-1.104E-4,- L2T01550
1      8.425E-4/ L2T01560
      DATA AGA4/-1.282E-5,-9.236E-6,-3.964E-3,-2.154E-3,- L2T01570
1      1.615E-2/ L2T01580
      DATA AGA5/5.586E-8,4.720E-8,1.596E-5,8.512E-6,6.350E- L2T01590
1      5/ L2T01600
      DATA AGA6/5.124E-8,5.235E-8,1.600E-5,9.392E-6,6.705E- L2T01610
1      5/ L2T01620
      DATA AGA7/2.706E-10,4.379E-10,-1.396E-7,-6.982E-8,- L2T01630
1      4.721E-7/ L2T01640
      DATA AGA8/-4.963E-11,-5.728E-11,-1.125E-8,-5.657E-9,- L2T01650
1      4.113E-8/ L2T01660
C  CHANGE ACCURACY TO 3 DECIMAL PTS (PGM DATA LIMIT) L2T01670
      WAVE1=FLOAT(IFIX(1000.*WAVE1))/1000. L2T01680
      READ (IOIN,1100) LNAME1,LNAME3,LNAME2,LNAME4,PH20,T,LZPATH L2T01690
      IF(IGEOSW.NE.1)GO TO 99 L2T01700
      LZPATH=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+ L2T01710
      +(PTS(6)-PTS(3))**2) L2T01720
      99 CONTINUE L2T01730
C  CHANGE UNITS - MB TO TORRI C TO K L2T01740
      PH20=PH20/TORRMB L2T01750
      T=T+CDEGK L2T01760
      IF (ICLMAT.EQ.1) T=TEMP+CDEGK L2T01770
      IF (ICLMAT.EQ.1) PH20=6.11*10.** L2T01780
1      <7.5*TEMP/(TEMP+237.3)*RH/<100.*TORRMB> L2T01790
C  PRINT HEADER WHEN THE WAVELENGTH CHANGES L2T01800
      IF (OLDWAY.NE.WAVEL) WRITE (IOOUT,1000) L2T01810
      OLDWAY=WAVEL L2T01820
      ABCOEF=0. L2T01830
      IF (WAVEL.EQ.0.0) GO TO 100 L2T01840
      IF (WAVEL.LT.0.8.OR.WAVEL.GT.11.0) GO TO 900 L2T01850
100  CALL LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID) L2T01860
      IF (LID.EQ.0) RETURN L2T01870
      P2=PH20*PH20 L2T01880
      IF (T.GE.260.AND.T.LE.320.AND.PH20.GE.0.AND. L2T01890
1      PH20.LE.35) GO TO 200 L2T01900
C  PRINT WARNING THAT TEMP OR PRESSURE IS OUT OF RANGE FOR L2T01910
C  ACCURATE CALCULATIONS AND CONTINUE. L2T01920
      WRITE (IOOUT,1300) L2T01930
200  IF (LID.GT.100) GO TO 500 L2T01940
      IF (LID.LT.1.0R.LID.GT.2) GO TO 900 L2T01950
      IF (LID-2)>300,400,900 L2T01960
C  ND:YAG LASER. NO MOLECULAR ABSORPTION AT 1.06 MICRONS. L2T01970
300  GO TO 800 L2T01980
C  CO2 LASER LINE P<20> L2T02000
400  T2=T*T L2T02010
      ABCOEF=AC020+AC021*T+AC022*PH20+AC023*T*PH20+AC024*P2+ L2T02020
1      AC025*T*P2+AC026*T2+AC027*T2*PH20+AC028*T2*P2 L2T02030
      GO TO 800 L2T02040
500  IF (LID.GT.200) GO TO 600 L2T02050
C  DF LASER. I IS THE LASER LINE INDEX L2T02060
      I=LID-100 L2T02070
      IF (I.GT.27) GO TO 900 L2T02080
      ABCOEF=ADF0(I)+ADF1(I)*T+ADF2(I)*PH20+ADF3(I)*T*PH20+ L2T02090
      ABCOEF=ADF0(I)+ADF1(I)*T+ADF2(I)*PH20+ADF3(I)*T*PH20+ L2T02100

```

```

        ADF4(I)*P2+ADF5(I)*T*P2          LZT02110
1      GO TO 800                         LZT02120
C 600 IF (LID.GT.300) GO TO 700         LZT02130
C CO LASER. I IS THE LASER LINE INDEX.   LZT02140
I=LID-200                                LZT02150
IF (I.GT.19) GO TO 900                  LZT02160
ABCOEF=AC00(I)+AC01(I)*T+AC02(I)*PH20+AC03(I)*T*PH20+
1     AC04(I)*P2+AC05(I)*T*P2          LZT02180
GO TO 800                                LZT02190
C 700 IF (LID.GT.400) GO TO 900         LZT02200
C GA AS LASER. I IS WAVELENGTH INDEX.    LZT02210
I=LID-300                                LZT02220
IF (I.GT.5) GO TO 900                  LZT02230
T2=T*T                                    LZT02240
ABCOEF=AGA0(I)+AGA1(I)*T+AGA2(I)*PH20+AGA3(I)*T*PH20+
1     AGA4(I)*P2+AGA5(I)*T*P2+AGA6(I)*T2+AGA7(I)*T2*
2     PH20+AGA8(I)*T2*P2              LZT02250
GO TO 800                                LZT02260
C NORMAL RETURN                          LZT02270
C 800 IF (ABCOEF.LT.0.) ABCOEF=0.        LZT02280
C COMPUTE TRANSMISSION                   LZT02290
LAZTRN=EXP(-LZPATH*ABCOEF)               LZT02300
WRITE (IOOUT,1200) WAVE,PH20,T,ABCOEF,LNAME1,LNAME3,
+LNAME2,LNAME4,LZPATH,LAZTRN             LZT02310
RETURN                                   LZT02320
C ERROR RETURN                          LZT02330
C 900 WRITE (IOOUT,1400)
LNAME1=BLANK                            LZT02340
LNAME3=BLANK                            LZT02350
LNAME2=BLANK                            LZT02360
LNAME4=BLANK                            LZT02370
LAZTRN=1.                               LZT02380
IERR=1                                  LZT02390
RETURN

C 1000 FORMAT (/,69X,10HABSORPTION,/,23X,11H WAVELENGTH,4X,
1     12HH20 PRESSURE,4X,11HTEMPERATURE,4X,
2     11HCOEFFICIENT,6X,4HLINE,9X,10HPATHLENGTH,4X,
3     12HTRANSMISSION,/,24X,9H(MICRONS),8X,6H(TORR),/)
4     11X,5H(ABS),10X,6H(KM-1),24X,4H(KM),/)           LZT02400
1100 FORMAT (4(A2),3F10.3)                LZT02410
1200 FORMAT (1H,22X,F09.3,F15.3,F16.2,E16.3,7X,4(A2),5X,
1     E10.4,5X,E10.4)                    LZT02420
1300 FORMAT (39H *** WARNING VALUE OF T OR PH20 OUT OF
1     10HRANGE *** /28H T RANGE = 260 TO 320 K ,
2     25HPH20 RANGE = 0 TO 35 TORR)       LZT02430
1400 FORMAT (40H *** ERROR WAVELENGTH OUT OF ACCEPTABLE
1     7HRANGE,26H .8 TO 11.0 MICRONS *** /,
2     37H CONTROL RETURNED TO MAIN FROM LZTRAN)        LZT02440
END                                     LZT02450
                                         LZT02460
                                         LZT02470
                                         LZT02480
                                         LZT02490
                                         LZT02500
                                         LZT02510
                                         LZT02520
                                         LZT02530
                                         LZT02540
                                         LZT02550
                                         LZT02560
                                         LZT02570
                                         LZT02580

```

```

SUBROUTINE LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)
C*****+
C THIS SUBROUTINE CONVERTS THE WAVELENGTH IN MICRONS OR THE LASER
C LINE NAME, IF WAVEL = 0, TO AN INTEGER LASER ID NUMBER WHICH IS
C USED BY LZTRAN.
C   ++ WAVEL TO LID, WHEN WAVEL NOT = 0 ++
C THE SUBROUTINE DOES A BINARY SEARCH OF THE ARRAY AWAVEL TO FIND
C A MATCH, WAVEL = AWAVEL(K). WHEN A MATCH IS FOUND THE LASER ID
C IS LOADED FROM THE ID ARRAY, LID = AID(K). THE LASER LINE NAME
C IS ALSO LOADED INTO TWO VARIABLES, LNAME1 = INAME1(K) LNAME2 =
C INAME2(K). IF AN EXACT MATCH IS NOT FOUND THE CLOSEST STANDARD
C TO THE PARAMETER WAVEL IS USED. A WARNING IS PRINTED AND K IS
C SET TO THE PROPER VALUE SO IT CAN BE USED TO INDEX THE ID AND
C LINE ARRAYS.
C   ++ LASER LINE TO LID, WHEN WAVEL = 0 ++
C WHEN THE WAVELENGTH PARAMETER IS ZERO THE CONVERSION IS DONE
C FROM LASER LINE NAME TO LID. A SEQUENTIAL SEARCH OF THE LINE
C NAME ARRAY IS PERFORMED. WHEN A MATCH IS FOUND K IS SET AND
C WAVEL AND LID ARE LOADED FROM THE APPROPRIATE ARRAYS. WHEN
C NO MATCH IS FOUND AN ERROR MESSAGE IS PRINTED AND LID IS SET
C TO ZERO. LID IS USED TO NOTIFY LZTRAN THAT AN ERROR HAS
C OCCURED AND NO CALCULATIONS SHOULD BE PERFORMED.
C   ++ PARAMETERS ++
WAVEL      LASER WAVELENGTH (MICRONS)
*** INPUTS IF WAVEL = 0.0 ***
LNAME1  FIRST 4 CHARS OF LASER LINE
LNAME2  SECOND 4 CHARS OF LASER LINE
++ RESULTS ++
LID      LASER LINE IDENTIFIER
LNAME1  FIRST 4 CHARS OF LASER LINE
LNAME2  SECOND 4 CHARS OF LASER LINE
C*****+
INTEGER AID(53)
COMMON /IOUNIT/ IOUNIT, IJOIN, IOUNIT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU
DIMENSION AWAVEL(53), INAME1(106), INAME2(53)
DATA AWAVEL/.85,.875,.9,
1  .925,.95,1.06,3.521,3.55,3.581,3.612,3.636,3.645,
2  3.666,3.679,3.698,3.715,3.731,3.752,3.765,3.8,
3  3.82,3.837,3.854,3.875,3.89,3.915,3.927,3.956,
4  3.965,3.999,4.005,4.046,4.089,4.908,4.918,4.928,
5  4.938,4.948,4.972,4.982,4.992,5.002,5.012,5.022,
6  5.032,5.043,5.047,5.054,5.057,5.067,5.078,5.088,
7  5.091/
DATA AID/301,302,303,304,305,1,127,126,125,124,123,
1  122,121,120,119,118,117,116,115,114,113,112,111,
2  110,109,108,107,106,105,104,103,102,101,219,218,
3  217,216,215,214,213,212,211,210,209,208,207,205,
4  206,204,203,202,201,2/
DATA INAME1/2HGA,2H,8,2HGA,2H,8,2HGA,2H,9,2HGA,2H,9,2HGA,2H,9,
1  2HHRU,2HBY,2HP1,2H<3,2HP1,2H<4,2HP1,2H<5,2HP1,2H<6,2HP2,2H<3,
2  2HP1,2H<7,2HP2,2H<4,2HP1,2H<8,2HP2,2H<5,2HP1,2H<9,2HP2,2H<6,
3  2HP1,2H<1,2HP2,2H<7,2HP2,2H<8,2HP3,2H<5,2HP2,2H<9,2HP3,2H<6,
4  2HP2,2H<1,2HP3,2H<7,2HP2,2H<1,2HP3,2H<8,2HP2,2H<1,2HP3,2H<9,
5  2HP2,2H<1,2HP3,2H<1,2HP3,2H<1,2HP4,2H<7,2HP4,2H<8,
6  2HP4,2H<9,2HP4,2H<1,2HP4,2H<1,2HP5,2H<7,2HP5,2H<8,2HP5,2H<9,
7  2HP5,2H<1,2HP5,2H<1,2HP5,2H<1,2HP5,2H<1,2HP5,2H<1,2HP6,2H<8,
8  2HP5,2H<1,2HP6,2H<9,2HP6,2H<1,2HP6,2H<1,2HP6,2H<1,2HP6,2H<20/
DATA INAME2/1H5,2H75,1H,2H25,1H5,1H,11*1H>,2H0>,5*
1  1H>,2H0>,1H>,2H1>,1H>,2H2>,1H>,2H3>,2H0>,2H1>
2  2H2>,3*1H>,2H0>,2H1>,3*1H>,2H0>,2H1>,2H2>,2H3>

```

```

3      2H4),1H),2H5),1H),2H0),2H1),2H2),1H)/          LZI00690
      DATA IFIRST,ILAST/1,53/
      I=IFIRST
      J=ILAST
      TWAVEL=WAVEL
C   CHECK FOR WAVELENGTH OR LASER LINE PASSED AS INPUT.  IF      LZI00700
C   WAVEL = 0 THEN DO A SEQUENTIAL SEARCH ON LINE NAME,      LZI00710
C   LNAME1,LNAME2.      LZI00720
      IF (<WAVEL,NE,0,0>) GO TO 200      LZI00730
C   INPUT = LASER LINE NAME      LZI00740
C   SEQUENTIAL SEARCH LOOP      LZI00750
      DO 100 K=IFIRST,ILAST      LZI00760
      KK=2*K-1
      IF(<(LNAME1,EQ,INAME1(KK)),AND,(LNAME3,EQ,INAME1(KK+1))>) GO TO 10      LZI00770
      GO TO 100      LZI00780
100  IF(<(LNAME2,EQ,INAME2(K))>) GO TO 600      LZI00790
C   CONTINUE      LZI00800
C   ERROR, NO MATCH ON LASER NAME      LZI00820
C   LID=0      LZI00830
C   PRINT ERROR MESSAGE      LZI00840
      WRITE (I00UT,900) LNAME1,LNAME3,LNAME2,LNAME4      LZI00850
      GO TO 700      LZI00860
C   INPUT = WAVELENGTH      LZI00870
C   BINARY SEARCH LOOP      LZI00880
200  K=<I+J>/2      LZI00890
      IF (<WAVEL,LE,AWAVEL(K)>) J=K-1      LZI00900
      IF (<WAVEL,GE,AWAVEL(K)>) I=K+1      LZI00910
      IF (<I,LE,J>) GO TO 200      LZI00920
C   DID WAVELENGTH MATCH A STANDARD IN AWAVEL(*) ?      LZI00930
      IF (<I-1,GT,J>) GO TO 600      LZI00940
C   WAVELENGTH NOT EXACTLY EQUAL TO ONE OF THE STANDARDS IN ARRAY      LZI00950
C   AWAVEL.  CHANGE WAVELENGTH TO EQUAL THE STANDARD IT IS CLOSEST TO.      LZI00960
C   THEN PRINT WARNING OF CHANGE.      LZI00970
      IF (<WAVEL,GT,AWAVEL(K),AND,K,LE,ILAST>) GO TO 500      LZI00980
      IF (<WAVEL,LT,AWAVEL(K),AND,K,LE,IFIRST>) GO TO 500      LZI00990
      IF (<WAVEL-AWAVEL(K)>) 300,600,400      LZI01000
C   WAVELET LT AWAVEL(K)      LZI01010
C   CHECK IF CLOSER TO AWAVEL(K) OR AWAVEL(K-1)      LZI01020
300  DELTA1=AWAVEL-AWAVEL(K-1)      LZI01030
      DELTA2=AWAVEL(K)-WAVEL      LZI01040
      IF (<DELTAL,T,DELTA2>) K=K-1      LZI01050
      GO TO 500      LZI01060
C   WAVELET GT AWAVEL(K)      LZI01070
C   CHECK IF WAVELET CLOSER TO AWAVEL(K) OR AWAVEL(K+1)      LZI01080
400  DELTA1=AWAVEL-AWAVEL(K)      LZI01090
      DELTA2=AWAVEL(K+1)-WAVEL      LZI01100
      IF (<DELTAL,T,DELTA2>) K=K+1      LZI01110
C   PRINT WARNING      LZI01120
500  WAVEL=AWAVEL(K)      LZI01130
      WRITE (I00UT,800) TWAVEL,WAVEL      LZI01140
C   LOAD LASER ID NUMBER      LZI01150
600  LID=AID(K)      LZI01160
C   LOAD LINE NAME      LZI01170
      KK=2*K-1      LZI01180
      LNAME1=INAME1(KK)
      LNAME3=INAME1(KK+1)
      LNAME2=INAME2(K)      LZI01190
C   LOAD WAVELENGTH      LZI01200
      WAVEL=AWAVEL(K)
700  RETURN      LZI01210
C
800  FORMAT (29H *** WARNING INPUT WAVELENGTH,F7.3,
1           11H CHANGED TO,F7.3,18H NEAREST STANDARD ,
2           14HWAVELENGTH ***)
900  FORMAT (24H *** ERROR LASER LINE #,4(A2),6H# NOT,
1           9HVALID ***,/26H CONTROL RETURNED TO MAIN,
2           12HFROM LZTRAN.)      LZI01220
      END      LZI01230
                                         LZI01240
                                         LZI01250
                                         LZI01260
                                         LZI01270
                                         LZI01280
                                         LZI01290
                                         LZI01300
                                         LZI01310
                                         LZI01320

```

```

SUBROUTINE DRTRAN(WAVE1,ICLMAT,TRNL0S,IERR) DRT00010
***** DRT00020
PURPOSE DRT00030
DIRTRAN-2 EXPLOSION PRODUCED AND VEHICLE GENERATED DUST MODEL DRT00040
INPUT,OUTPUT AND CALLING PROGRAM DRT00050
INPUTS DRT00060
VALUES IN ARGUMENT LIST DRT00070
ICLMAT INTEGER VALUE USED TO INDICATE HOW METEORLOGICAL DATA IS TO DRT00080
BE MADE AVAILABLE IF ICLMAT IS DRT00090
0 - MET1 IDENTIFIER WITH THE APPROPRIATE PARAMETERS ARE TO DRT00100
BE READ IN DRT00110
1 - NECESSARY METEORLOGICAL DATA IS PASSED IN COMMON/CLYMAT/ DRT00120
AND MET1 IS NOT TO BE READ IN DRT00130
DRT00140
DRT00150
DRT00160
DRT00170
WAVE1 WAVELENGTH IN MICROMETERS. USED TO DETERMINE NWL DRT00180
WHERE NWL IS AN INTEGER INDEX FOR WAVELENGTH DETERMINED DRT00190
WITHIN THE CODE DRT00200
NWL WAVE1 DRT00210
1 FOR 0.4 - 0.7 MICROMETER (VISIBLE) DRT00220
2 FOR 0.8 - 1.1 MICROMETER DRT00230
3 FOR 3.5 - 4.0 MICROMETER DRT00240
4 FOR 8.5 - 12.0 MICROMETER DRT00250
5 FOR 2100 - 3200 MICROMETER DRT00260
DRT00270
DRT00280
DRT00290
DRT00300
DRT00310
DRT00320
DRT00330
DRT00340
DRT00350
THE INPUT FILE MAY CONTAIN SEVERAL SEQUENCES OF THE FOLLOWING DRT00360
RECORDS. EACH SEQUENCE SEPARATED BY A GO CARD. ONCE THE INITIAL DRT00370
SEQUENCE HAS BEEN READ IN AND THE MINIMUM REQUIREMENTS FOR EXECUTION DRT00380
OF THE DESIRED OPTION HAS BEEN SATISFIED, ANY FOLLOWING SEQUENCE MAY DRT00390
CONTAIN A SUBSET OF THE INITIAL RECORDS REDEFINING INPUT VARIABLES DRT00400
AS DESIRED OR MAY CONTAIN A COMPLETELY NEW SET OF RECORDS. DRT00410
DRT00420
DRT00430
DRT00440
DRT00450
DRT00460
DRT00470
DRT00480
DRT00490
DRT00500
DRT00510
DRT00520
DRT00530
DRT00540
DRT00550
DRT00560
DRT00570
DRT00580
DRT00590
DRT00600
DRT00610
DRT00620
DRT00630
DRT00640
DRT00650
DRT00660
DRT00670
DRT00680
DRT00690
INPUTS TO BE READ
EACH INPUT RECORD BEGINS WITH A 4 LETTER IDENTIFIER IN COLUMNS
1-4 FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, UP TO 9, 8 COLUMNS
PER FIELD BEGINNING IN COLUMN 9.
THE INPUT FILE MAY CONTAIN SEVERAL SEQUENCES OF THE FOLLOWING
RECORDS. EACH SEQUENCE SEPARATED BY A GO CARD. ONCE THE INITIAL
SEQUENCE HAS BEEN READ IN AND THE MINIMUM REQUIREMENTS FOR EXECUTION
OF THE DESIRED OPTION HAS BEEN SATISFIED, ANY FOLLOWING SEQUENCE MAY
CONTAIN A SUBSET OF THE INITIAL RECORDS REDEFINING INPUT VARIABLES
AS DESIRED OR MAY CONTAIN A COMPLETELY NEW SET OF RECORDS.
** EACH SET OF INPUTS MUST END WITH A DONE CARD**
RECORD 1
MET1
NATMOS INTEGER WITH VALUES 1 TO 6 CORRESPONDING TO PASQUILL DRT00450
CATEGORIES A TO F. DRT00460
DRT00470
ZTMP THE HEIGHT AT WHICH A TEMPERATURE MEASUREMENT IS DRT00480
AVAILABLE. VALID RANGE 0.5 - 100.0 M. DRT00490
DRT00500
TMPMES THE TEMPERATURE MEASURED IN DEGREES KELVIN TAKEN AT DRT00510
HEIGHT ZTMP. VALID RANGE 270.0 - 315.0. DRT00520
DRT00530
ZWND THE HEIGHT AT WHICH A WIND SPEED MEASUREMENT IS DRT00540
AVAILABLE. VALID RANGE 0.5 - 100.0 M. DRT00550
DRT00560
WNDMES THE WIND SPEED IN METERS/SECOND MEASURED AT ZWND DRT00570
VALID RANGE .1 - 20.0 M/S DRT00580
DRT00590
THWND THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES DRT00600
WITH THE USER'S POSITIVE X AXIS MEASURED IN DEGREES DRT00610
COUNTERCLOCKWISE. WHERE THE USERS POSITIVE X-AXIS DRT00620
POINTS EAST. THUS THWND IS THE ANGLE THAT THE WIND DRT00630
VELOCITY VECTOR MAKES WITH THE EAST. DRT00640
VALID RANGE -360.0 - 360.0 DEGREES. DRT00650
NOTE: THWND IS NOT NEEDED FOR OPTION 3 DRT00660
DRT00670
DRT00680
DRT00690

```

RECORD 2		DRT00700
MET2		DRT00710
ID	A FLAG TO INDICATE WHETHER THE INVERSION LAYER HEIGHT IS GROWING OR NOT. IF ID IS	DRT00720
	0. THE INVERSION LAYER HEIGHT IS RELATIVELY CONSTANT	DRT00730
	1. THE INVERSION LAYER HEIGHT IS GROWING	DRT00740
PHI	THE LATITUDE OF THE DETONATION SITE, VALID RANGE 1.0 - 90. DEGREES. THAT IS THE NORTHERN HEMISPHERE.	DRT00750
RECORD 3		DRT00760
SU1L		DRT00770
NSOIL	INTEGER INDEX OF SOIL TYPE. NSOIL IS	DRT00780
	1: FOR SOIL-1, <DATA GRAF-II> EXPLOSIONS ONLY.	DRT00790
	2: FOR SOIL-2, <DATA DIRT-I> EXPLOSIONS ONLY.	DRT00800
	3: FOR SOIL-3, <DATA SMOKEWEEK-II> VEHICLES ONLY.	DRT00810
DSOD	DEPTH OF SOD IN METERS VALID RANGE: 0.0 - 1.0 M. NOTE: FOR VEHICLE MODEL IF DSOD>0.0 NO DUST IS GENERATED	DRT00820
SILT	SILT CONTENT OF SOIL (PARTICLE DIAMETERS < 75 MICRONS) I.E. SILT=.15 INDICATES A SILT CONTENT OF 15% NOTE: THIS INPUT NEEDED ONLY FOR VEHICLE MODEL (IOPT=5)	DRT00830
RECORD 4		DRT00840
CHAR		DRT00850
NCHRG	CHARGE TYPE INDEX WITH FOLLOWING VALUES	DRT00860
	1. SURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC TIP ON GROUND	DRT01030
	2. BARE CHARGE ON SURFACE	DRT01040
	3. 30 DEGREE TILTED TIP AT 0.3 METER DEPTH	DRT01050
	4. 30 DEGREE TILTED TIP AT 0.6 METER DEPTH	DRT01060
	5. HORIZONTAL PROJECTILE ON SURFACE	DRT01070
	DEFAULT VALUE IS 1 IF NCHRG IS NOT BETWEEN 1 AND 5.	DRT01080
CHWT	THE WEIGHT OF THE CHARGE IN KG-TNT. VALID RANGE: 0.1 - 100.0 KG-TNT.	DRT01090
DETDEP	THE DEPTH OF DETONATION IN METERS. VALID RANGE: 0.0 - 2.0 M.	DRT01100
RECORD 5		DRT01110
EXPL		DRT01120
NARY	TYPE OF CHARGE DISTRIBUTION (USED FOR PROPER INPUT AND OUTPUT FORMATS) IF THE VALUE OF NARY AND IOPT OF THE GO CARD ARE NOT COMPATIBLE CATASTROPHE COULD RESULT! NOTE: WHEN NARY IS	DRT01130
	1: IOPT MUST ALSO BE 1.	DRT01140
	2: IOPT MUST ALSO BE 2.	DRT01150
	3: IOPT MUST BE 4.	DRT01160
	1.-SIMULTANEOUS BURST,UNIFORMLY DISTRIBUTED CHARGES IN A PARALLELOGRAM. (SPECIAL CASES ARE ,SINGLE CHARGE ,RECTANGLE AND ZIG ZAG PATTERN)	DRT01170
	2.-SIMULTANEOUS BURST,RANDOMLY DISTRIBUTED CHARGES.	DRT01180
	3.-SEQUENTIAL IN TIME AND RANDOM IN SPACE DISTRIBUTION OF CHARGES.	DRT01190
	NOTE: WHEN NARY=2. EACH CHARGE LOCATION MUST BE SPECIFIED	DRT01200
		DRT01210
		DRT01220
		DRT01230
		DRT01240
		DRT01250
		DRT01260
		DRT01270
		DRT01280
		DRT01290
		DRT01300
		DRT01310
		DRT01320
		DRT01330
		DRT01340
		DRT01350
		DRT01360
		DRT01370
		DRT01380
		DRT01390

	WHEN NARY=3, EACH CHARGE LOCATION AND DETONATION TIME	DRT01400
	MUST BE SPECIFIED	DRT01410
	THE CHARGE LOCATIONS ARE INPUT DIRECTLY FOLLOWING	DRT01420
	THIS INPUT RECORD WITH ONE LOCATION (AND DETONATION	DRT01430
	TIME IF APPROPRIATE) PER RECORD.	DRT01440
NCHS	SINGLY DIMENSIONED ARRAY SPECIFYING NUMBER OF CHARGES WITH THE MAXIMUM TOTAL OF CHARGES 200. WHEN NARY=1, NCHS(1) IS THE NUMBER OF CHARGES IN THE DIRECTION OF SIDE1 AND NCHS(2) IS THE NUMBER OF CHARGES IN THE DIRECTION OF SIDE2. FOR A SINGLE CHARGE SET NCHS(1)=NCHS(2)=1. WHEN NARY=2, OR 3, NCHS(1) IS THE TOTAL NUMBER OF CHARGES AND SET NCHS(2)=1..	DRT01450 DRT01460 DRT01470 DRT01480 DRT01490 DRT01500 DRT01510 DRT01520 DRT01530 DRT01540
SRCBAS	SINGLY DIMENSIONED ARRAY CONTAINING THE COORDINATES OF A CORNER POINT OF THE BOUNDING PARALLELOGRAM WHEN NARY=1. AND IS ALSO USED AS THE REFERENCE CHARGE BY THE OBSERVER. THAT IS, SRCBAS IS THE ORIGIN OF THE OBSERVER'S COORDINATE SYSTEM. WHEN NARY=2, OR 3, SRCBAS SHOULD NOT APPEAR ON THE INPUT FILE AS COOR(I,1) IS USED AS THE REFERENCE CHARGE. WHERE COOR(I,1) IS THE FIRST CHARGE LOCATION SPECIFIED. VALID RANGE -10000.0 - 10000.0	DRT01550 DRT01560 DRT01570 DRT01580 DRT01590 DRT01600 DRT01610 DRT01620 DRT01630 DRT01640
SIDE1	SINGLY DIMENSIONED ARRAY NECESSARY ONLY WHEN NARY=1., SPECIFYING ONE SIDE OF THE BOUNDING PARALLELOGRAM FROM THE POINT SRCBAS(I). THAT IS, SIDE1 IS A VECTOR TO THE NEXT CHARGE ALONG ONE SIDE OF THE PARALLELOGRAM. WHEN NARY=2, OR 3. THIS VARIABLE SHOULD NOT APPEAR ON THE INPUT FILE.	DRT01650 DRT01660 DRT01670 DRT01680 DRT01690 DRT01700 DRT01710
SIDE2	SINGLY DIMENSIONED ARRAY NECESSARY ONLY WHEN NARY=1. SPECIFYING A SECOND SIDE OF THE BOUNDING PARALLELOGRAM FROM THE POINT SRCBAS(I). THAT IS SIDE2 IS A VECTOR TO TO THE NEXT CHARGE ALONG THE SECOND SIDE OF THE PARALLELOGRAM. WHEN NARY=2, OR 3. THIS VARIABLE SHOULD NOT APPEAR ON THE INPUT FILE.	DRT01720 DRT01730 DRT01740 DRT01750 DRT01760 DRT01770 DRT01780 DRT01790
**	THE FOLLOWING RECORD MUST APPEAR THE APPROPRIATE NUMBER OF TIMES	DRT01800
**	IMMEDIATELY FOLLOWING THE ABOVE RECORD IF NARY IS 2, OR 3. ON THE	DRT01810
**	ABOVE RECORD. THAT IS IT MUST APPEAR THE SAME NUMBER OF TIMES	DRT01820
**	AS THERE ARE CHARGES AS SPECIFIED ON RECORD EXPL. (I.E. IF NARY=2, ** AND NCHS=5, THE THIS RECORD MUST APPEAR 5 TIMES.	DRT01830 DRT01840 DRT01850 DRT01860 DRT01870 DRT01880 DRT01890 DRT01900 DRT01910 DRT01920 DRT01930 DRT01940 DRT01950 DRT01960 DRT01970 DRT01980 DRT01990 DRT02000 DRT02010 DRT02020 DRT02030 DRT02040 DRT02050 DRT02060 DRT02070 DRT02080 DRT02090
RECORD 6 LNUCA		
COOR	DOUBLY DIMENSIONED ARRAY CONTAINING THE DETONATION COORDINATES FOR EACH CHARGE WHEN NARY=2, OR 3.. WHEN NARY=1. THIS VARIABLE NEED NOT BE SPECIFIED AS THE CHARGE LOCATIONS ARE CALCULATED IN THE CODE FROM NCHS,SIDE1,SIDE2.	DRT01900 DRT01910 DRT01920 DRT01930
TSTAG	SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF DETONATION OF EACH CHARGE. THIS IS ONLY SPECIFIED WHEN NARY=3..	DRT01940 DRT01950 DRT01960 DRT01970 DRT01980 DRT01990 DRT02000 DRT02010 DRT02020 DRT02030 DRT02040 DRT02050 DRT02060 DRT02070 DRT02080 DRT02090
RECORD 7 VEHC		
V0	DOUBLY DIMENSIONED ARRAY CONTAINING THE INITIAL POSITION OF THE VEHICLE. V0(1)=X-COORDINATE V0(2)=Y-COORDINATE VALID RANGE: -10000.0 - 10000.0	DRT02030 DRT02040 DRT02050
VEHDIR	VEHICLE DIRECTION. THE ANGLE THAT THE VEHICLE VELOCITY VECTOR MAKES WITH THE USER'S POSITIVE X-AXIS MEASURED IN DEGREES COUNTERCLOCKWISE. WHERE THE USER'S POSITIVE	DRT02060 DRT02070 DRT02080 DRT02090

	X-AXIS POINTS EAST. THUS VEHDIR IS THE ANGLE THE VELOCITY VECTOR MAKES WITH THE EAST. VALID RANGE: -360.0 - 360.0	DRT02100 DRT02110 DRT02120 DRT02130 DRT02140 DRT02150 DRT02160 DRT02170 DRT02180 DRT02190 DRT02200 DRT02210 DRT02220 DRT02230 DRT02240 DRT02250 DRT02260 DRT02270 DRT02280 DRT02290 DRT02300 DRT02310 DRT02320 DRT02330
VEHSPD	VEHICLE SPEED IN M/S	DRT02340
VEHWID	VEHICLE WIDTH IN METERS	DRT02350
VEHWHT	VEHICLE WEIGHT IN KGS.	DRT02360
VENTYP	TRACTION MECHANISM =0. VEHICLE HAS TIRES =1. VEHICLE IS TRACKED	DRT02370 DRT02380 DRT02390 DRT02400 DRT02410 DRT02420 DRT02430 DRT02440 DRT02450 DRT02460
RECORD 8		
TRNC		
TRNCOR	A SIMPLY DIMENSIONED ARRAY CONTAINING THE THREE COORDINATES OF THE TRANSMITTER. THE COORDINATE SYSTEM MUST BE IN METERS. THE THIRD COORDINATE IS RESTRICTED TO BE BETWEEN .5 AND 10000.0 METERS (HEIGHT). VALID RANGE OF THE FIRST TWO COORDINATES: -10000.0 - 10000.0 M.	DRT02340 DRT02350 DRT02360 DRT02370 DRT02380 DRT02390 DRT02400 DRT02410 DRT02420 DRT02430 DRT02440 DRT02450 DRT02460
** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE ** ARRAY TRNCOR NEED NOT BE SPECIFIED.		
TRNMIN	VALUE SUCH THAT A TRANSMITTANCE BELOW THIS VALUE CAN BE CONSIDERED ZERO. DEFAULT IS 1.E-05 VALID RANGE: 1.0 - 1.E-05	DRT02340 DRT02350 DRT02360 DRT02370 DRT02380 DRT02390 DRT02400 DRT02410 DRT02420 DRT02430 DRT02440 DRT02450 DRT02460
RECORD 9		
RECC		
RECCOR	A SIMPLY DIMENSIONED ARRAY CONTAINING THE THREE COORDINATES OF THE RECEIVER. (METERS) THE THIRD COORDINATE IS RESTRICTED TO BE BETWEEN .5 AND 10000.0 METERS. VALID RANGE OF THE FIRST TWO COORDINATES IS: -10000.0 - 10000.0 M.	DRT02420 DRT02430 DRT02440 DRT02450 DRT02460
** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE ** ARRAY RECCOR NEED NOT BE SPECIFIED.		
RECORD 10		
OBSC		
OBSCOR	A SIMPLY DIMENSIONED ARRAY CONTAINING THE X AND Y COORDINATES, RESP., OF THE OBSERVER. (METERS) VALID RANGE: -10000.0 - 10000.0	DRT02510 DRT02520 DRT02530 DRT02540
SPCHT	A SPECIFIED HEIGHT IN METERS AT WHICH THE WIDTH OF THE CLOUD AS VIEWED FROM POSITION OBSCOR IS DESIRED. MUST BE BETWEEN 1. AND 5. METERS.	DRT02550 DRT02560 DRT02570
** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE ** ARRAY OBSCOR AND VARIABLE SPCHT NEED NOT BE SPECIFIED.		
RECORD 11		
TIMS		
TSTART	TIME AFTER DETONATION TO START TRANSMITTANCE AND/OR CLOUD DIMENSION CALCULATIONS VALID RANGE: .5 - 1000.0 SEC.	DRT02580 DRT02590 DRT02600 DRT02610 DRT02620 DRT02630 DRT02640
TEND	TIME AFTER DETONATION TO TERMINATE TRANSMITTANCE AND/OR CLOUD DIMENSIONS. VALID RANGE: .5-1000.0 SEC. (TEND MUST BE .GE. TSTART)	DRT02650 DRT02660 DRT02670 DRT02680 DRT02690 DRT02700
TINC	TIME INCREMENT BETWEEN CALCULATIONS	DRT02650 DRT02660 DRT02670 DRT02680 DRT02690 DRT02700

RECORD 12	DRT02710
GO ** THIS CARD INDICATES THAT THIS SEQUENCE OF INPUTS ARE	DRT02720
COMPLETE AND CALCULATIONS ARE TO BEGIN.	DRT02730
IOPT OPTION TO BE USED	DRT02740
1. SIMULTANEOUS BURST ,UNIFORMLY DISTRIBUTED CHARGES IN A	DRT02750
PARALLELOGRAM	DRT02760
2. SIMUTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES	DRT02770
3. THE CODE IS TO PRECOMPUTE A SINGLE CLOUD AND STORE ON	DRT02780
AN EXTERNAL FILE FOR USE LATER	DRT02790
4. THE CODE IS TO USE A CLOUD THAT HAS BEEN PRECOMPUTED	DRT02800
(NO CLOUD DIMENSIONS ARE COMPUTED FOR THIS OPTION)	DRT02810
5. VEHICLE DUST MODEL	DRT02820
IFILE FORTRAN LOGICAL UNIT TO WHICH THE CODE IS TO WRITE FOR	DRT02830
OPTION 3 OTHERWISE IT NEED NOT BE SPECIFIED	DRT02840
RECORD 13	DRT02850
DONE ** THIS RECORD INDICATES THAT THE USER HAS COMPLETED HIS	DRT02860
DESIRED SEQUENCE OF INPUTS AND ALL CALCULATIONS ARE	DRT02870
TERMINATED	DRT02880
OUTPUTS	DRT02890
ZINV THE ESTIMATED INVERSION HEIGHT.	DRT02900
TRNLOS THE TRANSMITTANCE ALONG THE LINE OF SIGHT BETWEEN	DRT02910
THE TRANSMITTER AND THE RECEIVER.	DRT02920
IERR INTEGER ERROR CODE WHICH EQUALS 1 IF A FATAL ERROR	DRT02930
OCCURS AND 0 OTHERWISE	DRT02940
NERR INTEGER ERROR CODE WITH THE VALUES	DRT02950
0 NO ERRORS	DRT02960
4 NO TRANSMITTER AND RECEIVER OR OBSERVER	DRT02970
COORDINATES WERE SPECIFIED SO NO RESULTS WERE	DRT02980
CALCULATED.	DRT02990
7 THE CALCULATION OF ATMOSPHERIC PARAMETERS DID	DRT03000
NOT CONVERGE.	DRT03010
CNTRD A SIMPLY DIMENSIONED ARRAY CONTAINING THE HORIZONTAL	DRT03020
COORDINATE AND THE VERTICAL COORDINATE OF THE	DRT03030
CENTROID OF THE CLOUD.	DRT03040
HEIGHT THE HEIGHT OF THE CLOUD IN METERS.	DRT03050
CENWTH THE WIDTH OF THE CLOUD IN METERS AT THE CENTROID	DRT03060
HEIGHT	DRT03070
SPCWTH THE WIDTH OF THE CLOUD IN METERS AT THE SPECIFIED	DRT03080
HEIGHT	DRT03090
NCPTS THE NUMBER OF POINTS DETERMINED ON THE EDGE OF THE	DRT03100
CLOUD.	DRT03110
CPTS A DOUBLY DIMENSIONED ARRAY CONTAINING THE COORDINATES	DRT03120
OF POINTS ON THE EDGE OF THE CLOUD. CPTS(1, J)	DRT03130
IS THE HORIZONTAL COORDINATE OF THE J-TH POINT	DRT03140
AND CPTS(2, J)IS THE VERTICAL COORDINATE OF THE	DRT03150
J-TH POINT. THE FIRST INDEX MUST BE DIMENSIONED	DRT03160
TO 2.	DRT03170

C SUBROUTINES CALLED DRT03410
 C DUSTCL CONTROLLING ROUTINE FOR THE CALCULATION OF CLOUD DIMENSIONS DRT03420
 AND TRANSMITTANCES THROUGH DUST CLOUDS FOR OPTIONS 1 AND 2 DRT03430
 GIVEN METEORLOGICAL DATA, SOIL AND EXPLOSIVE DRT03440
 CHARACTERISTICS, AND WAVELENGTH. DRT03450
 DRT03460
 COMPCL CONTROLLING ROUTINE FOR PRECOMPUTING A SINGLE EXPLOSION DRT03470
 (OPTION 3) GIVEN METEORLOGICAL DATA, SOIL AND EXPLOSIVE DRT03480
 CHARACTERISTICS. ALSO USES THIS PRECOMPUTED CLOUD AT SOME DRT03490
 LATER RUNNING OF THE CODE (OPTION 4) TO ESTIMATE A DRT03500
 TRANSMITTANCE GIVEN TRANSMITTER AND RECEIVER COORDINATES. DRT03510
 DRT03520
 VEHCL CONTROLLING ROUTINE FOR THE CALCULATION OF A TRANSMITTANCE DRT03530
 THROUGH A VEHICLE GENERATED DUST CLOUD (OPTION 5) GIVEN DRT03540
 METEORLOGICAL DATA, SOIL CHARACTERISTICS, VEHICLE DRT03550
 CHARACTERISTICS, AND WAVELENGTH. DRT03560
 DRT03570
 DRT03580
 **** LOGICAL NEWATM, NEWSRC, LOSTRN, EDGE, NEWTIM, CLMRED, DHDT, ONCE DRTU3590
 LOGICAL TEST, NEWVEH, NEWCOR DRT03600
 LOGICAL M1, M2, SL, CH, EX, TC, RC, OC, TM, VH DRT03610
 INTEGER VEHTYP DRT03620
 REAL M, N DRT03630
 DIMENSION ZTMP(2), TMPMES(2), ZWND(2), WNDMES(2), TRNCOR(3) DRT03640
 DIMENSION SRCBAS(2), SIDE1(2), SIDE2(2), NCBS(2) DRT03650
 1, RECCOR(3), CPTS(2,6), CNTRD(2), OBSCOR(2) DRT03660
 DIMENSION RDIN(10), RKEY(12), Y0(2), PAS(6) DRT03670
 COMMON /IOUNIT/ IOUNT, IODOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUDRT03680
 COMMON /CLYMAT/ TEMP, PRESS, RH, AH, DP, VIS, CLDAMT, CLDHYT, DRT03690
 1 FOGPRB, WNDVEL, WINDIR, IPASCT DRT03700
 COMMON/MOS/ DIFF(2,200), NCHTOT, PRSEP(200), NTOT, NARY, ITOT, DRT03710
 + COOR(2,200), TSTAG(200), DMMY(401) DRT03720
 COMMON/WNDPRM/DXZ0, DYX0, DZ0, U0, M, N, ZINV DRT03730
 COMMON/TRANNS/ THRESH, TEST, NWL, NSOIL DRT03740
 COMMON/GEOM/COSTH2, SINTH, SINTH2, VISEXT, RTPI, SCRNC(2) DRT03750
 COMMON/GOMET/PTSC(15), IGEOSW DRT03760
 COMMON/OPTION/IOPt, IFILE DRT03770
 DATA RKEY/4HMET1, 4HMET2, 4HSOIL, 4HCHAR, 4HEXPL, 4HVEHC, 4HTRNC, DRT03780
 1 4HRECC, 4HOBSC, 4HTIMS, 4HGO , 4HDONE/ DRT03790
 DATA PAS/4HA DRT03800
 DATA M1, M2, SL, CH, EX, TC, RC, OC, TM, VH/, FALSE,, FALSE,, FALSE,, DRT03810
 1, FALSE,, FALSE,, FALSE,, FALSE,, FALSE,, FALSE,, FALSE,, DRT03820
 DATA NEWATM, NEWSRC, NEWVEH, LOSTRN, EDGE, NEWTIM/, FALSE,, FALSE,, DRT03830
 1, FALSE,, FALSE,, FALSE,, FALSE,, DRT03840
 DATA NEWCOR/, FALSE,/ DRT03850
 DATA VISEXT, RTPI/.1, 1.772454/ DRT03860
 IERR=0 DRT03870
 CLMRED=.FALSE. DRT03880
 DHDT=.FALSE. DRT03890
 ONCE=.FALSE. DRT03900
 TEST=.FALSE. DRT03910
 WRITE(IODOUT, 800) DRT03920
 800 FORMAT(1H0, 36X, 42HDIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION, DRT03930
 1 15H CALCULATION, //, 36X, 60H*** NOTE -- ALL UNITS ARE MKS UNLESS DRT03940
 2 OTHERWISE SPECIFIED ***/, //) DRT03950
 DO 5 K=1, 200 DRT03960
 TSTAG(K)=0.0 DRT03970
 5 CONTINUE DRT03980
 DRT03990
 C DETERMINE INTEGER INDEX FOR WAVELENGTH DRT04000
 10 IF(WAVE1.LT.0.4)GO TO 29 DRT04010
 IF(WAVE1.GT.0.7)GO TO 21 DRT04020
 NWL=1 DRT04030
 GO TO 30 DRT04040
 21 IF(WAVE1.LT.0.8)GO TO 29 DRT04050
 IF(WAVE1.GT.1.1)GO TO 22 DRT04060
 NWL=2 DRT04070
 GO TO 30 DRT04080
 DRT04090

```

22 IF(WAVE1.LT.3.5)GO TO 29          DRT04100
    IF(WAVE1.GT.4.0)GO TO 23          DRT04110
    NWL=3                            DRT04120
    GO TO 30                          DRT04130
23 IF(WAVE1.LT.8.5)GO TO 29          DRT04140
    IF(WAVE1.GT.12.0)GO TO 24          DRT04150
    NWL=4                            DRT04160
    GO TO 30                          DRT04170
24 IF(WAVE1.LT.2100.)GO TO 29          DRT04180
    IF(WAVE1.GT.3200.)GO TO 29          DRT04190
    NWL=5                            DRT04200
    GO TO 30                          DRT04210
29 WRITE(I00UT,802)                  DRT04220
802 FORMAT(37X,38H*** DIRTRAN ERROR - WAVE1 OUT OF RANGE)
    IERR=1                           DRT04230
    GO TO 999                         DRT04240
30 CONTINUE                         DRT04250
DRT04260
DRT04270
DRT04280
C READ DATA AND STORE APPROPRIATELY
C
    DO 300 II=1,15                  DRT04290
    IF(II.EQ.15)GO TO 900           DRT04300
    READ(I0IN,700)(RDIN(J),J=1,10)   DRT04310
700 FORMAT(A4,4X,9F8.2)             DRT04320
    IF(RDIN(1).EQ.RKEY(1))GO TO 50  DRT04330
    IF(RDIN(1).EQ.RKEY(2))GO TO 70  DRT04340
    IF(RDIN(1).EQ.RKEY(3))GO TO 90  DRT04350
    IF(RDIN(1).EQ.RKEY(4))GO TO 110 DRT04360
    IF(RDIN(1).EQ.RKEY(5))GO TO 130 DRT04370
    IF(RDIN(1).EQ.RKEY(6))GO TO 150 DRT04380
    IF(RDIN(1).EQ.RKEY(7))GO TO 170 DRT04390
    IF(RDIN(1).EQ.RKEY(8))GO TO 190 DRT04410
    IF(RDIN(1).EQ.RKEY(9))GO TO 210 DRT04420
    IF(RDIN(1).EQ.RKEY(10))GO TO 230 DRT04430
    IF(RDIN(1).EQ.RKEY(11))GO TO 310 DRT04440
    IF(RDIN(1).EQ.RKEY(12))GO TO 999 DRT04450
    WRITE(I00UT,804)                  DRT04460
804 FORMAT(33X,52H***DIRTRAN-2 ERROR, INPUT DOES NOT CONFORM TO PROPER DRT04470
    1        14H CONVENTION***)
    WRITE(I00UT,806)(RDIN(J),J=1,9)   DRT04480
806 FORMAT(26X,A4,4X,9F8.2)         DRT04490
    GO TO 999                         DRT04500
DRT04510
C STORE AND PRINT OUT ATMOSPHERIC CONDITIONS
C
50 CONTINUE                         DRT04520
    M1=.TRUE.                         DRT04530
    N10=1                            DRT04540
    IF(ICLMAT.EQ.1)GO TO 55          DRT04550
    NATMOS=IFIX(RDIN(2))             DRT04560
    ZTMP(1)=RDIN(3)                  DRT04570
    TMPMES(1)=RDIN(4)                DRT04580
    ZWND(1)=RDIN(5)                  DRT04590
    WNDMES(1)=RDIN(6)                DRT04600
    THWND=RDIN(7)                   DRT04610
    GO TO 60                          DRT04620
DRT04630
DRT04640
DRT04650
DRT04660
C IPSCAT PASQUILL CATEGORY
C
    WNDVEL  WIND VELOCITY IN M/S MEASURED AT 2 M. ABOVE GROUND DRT04670
    WINDIR  WIND DIRECTION IN DEGREES CLOCKWISE FROM TRUE NORTH DRT04680
    TEMP    TEMPERATURE IN DEGREES C MEASURED AT 2 M. ABOVE GROUND DRT04690
DRT04700
DRT04710
55 NATMOS=IPASCT                  DRT04720
    ZTMP(1)=2,                         DRT04730
    ZWND(1)=2,                         DRT04740
    WNDMES(1)=WNDVEL                 DRT04750
    TMPMES(1)=TEMP+273.0              DRT04760
    THWND=270.0-WINDIR                DRT04770
60 CONTINUE                         DRT04780
    WRITE(I00UT,808)                  DRT04790

```

```

808 FORMAT(1X)
     WRITE(I00UT,810)PAS(NATMOS)                                DRT04800
810 FORMAT(50X,28HPASQUILL CATEGORY                           DRT04810
     WRITE(I00UT,812)(ZTMP(I),TMPHES(I),ZWND(I),WNDMES(I),I=1,NIO),A4) DRT04820
812 FORMAT(36X,4H HT ,F8.2,7H TEMP ,F8.2,7H HT,F8.2,7H WIND ,      DRT04830
   ,F8.2)                                                       DRT04840
     WRITE(I00UT,814)THWND                                     DRT04850
814 FORMAT(51X,22H WIND DIRECTION ,F8.2)                      DRT04860
GO TO 300
70 CONTINUE
M2=.TRUE.
ID=IFIX(RDIN(2))
IFC ID,NE,0>DHDT=.TRUE.
PHI=RDIN(3)
WRITE(I00UT,819)PHI
IFC DHDT>WRITE(I00UT,816)
816 FORMAT(47X,37HTHE INVERSION LAYER HEIGHT IS GROWING)      DRT04950
IFC .NOT.DHDT>WRITE(I00UT,818)                                DRT04960
818 FORMAT(47X,38HTHE INVERSION LAYER HEIGHT IS CONSTANT)       DRT04970
819 FORMAT(/,52X,20HLATITUDE ,F8.2)                            DRT04980
GO TO 300
C STORE AND WRITE SOIL CHARACTERISTICS
C
90 CONTINUE
SL=.TRUE.
NSOIL=IFIX(RDIN(2))
DSOD=RDIN(3)
SILT=RDIN(4)
IFC NSOIL,EQ,1>WRITE(I00UT,821)                                DRT05000
820 FORMAT(56X,15HSILT CONTENT ,F5.2)                          DRT05010
821 FORMAT(/,63X,6HSODL-1)                                      DRT05020
IFC NSOIL,EQ,2>WRITE(I00UT,822)                                DRT05030
822 FORMAT(/,63X,6HSOIL-2)                                      DRT05040
IFC NSOIL,EQ,3>WRITE(I00UT,710)                                DRT05050
710 FORMAT(/,23X,8H SOIL-3)                                     DRT05060
IFC SILT.GT.1.E-06>WRITE(I00UT,820)SILT
WRITE(I00UT,823)DSOD
823 FORMAT(53X,21H DEPTH OF SOD ,F5.2)                         DRT05070
IFC NSOIL.LT.1.0R.NSOIL.GT.2>NSOIL=2                         DRT05080
GO TO 300
C STORE AND WRITE EXPLOSIVE CHARGE CHARACTERISTICS
C
110 CONTINUE
CH=.TRUE.
NCHRG=IFIX(RDIN(2))
IFC NCHRG,LT,1.0R.NCHRG,GT,5>NCHRG=1                         DRT05100
CHWT=RDIN(3)
DETDEP=RDIN(4)
IFC NCHRG,EQ,1>WRITE(I00UT,824)                                DRT05110
824 FORMAT(/,35X,47HSURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC,, DRT05120
   ,14H TIP ON GROUND)                                         DRT05130
IFC NCHRG,EQ,2>WRITE(I00UT,825)                                DRT05140
825 FORMAT(/,55X,22HBARE CHARGE ON SURFACE)                   DRT05150
IFC NCHRG,EQ,3>WRITE(I00UT,826)                                DRT05160
826 FORMAT(/,46X,39H30 DEGREE TILTED TIP AT 0.3 METER DEPTH)  DRT05170
IFC NCHRG,EQ,4>WRITE(I00UT,827)                                DRT05180
827 FORMAT(/,46X,39H30 DEGREE TILTED TIP AT 0.6 METER DEPTH)  DRT05190
IFC NCHRG,EQ,5>WRITE(I00UT,828)                                DRT05200
828 FORMAT(/,50X,32HHORIZONTAL PROJECTILE ON SURFACE)        DRT05210
WRITE(I00UT,829)CHWT
829 FORMAT(45X,30HWEIGHT OF CHARGE ,F8.2,4H KG.)              DRT05220
WRITE(I00UT,830)DETDEP
830 FORMAT(47X,30HDETONATION DEPTH ,F8.2)                      DRT05230
GO TO 300
C STORE AND WRITE OUT INFORMATION ABOUT THE DETONATION LOCATIONS
C
130 CONTINUE

```

```

EX=.TRUE.
NARY=IFIX(RDIN(2))
NCHS(1)=IFIX(RDIN(3))
NCHS(2)=IFIX(RDIN(4))
SRCBAS(1)=RDIN(5)
SRCBAS(2)=RDIN(6)
SIDE1(1)=RDIN(7)
SIDE1(2)=RDIN(8)
SIDE2(1)=RDIN(9)
SIDE2(2)=RDIN(10)
IF(NARY.EQ.2)GO TO 133
IF(NARY.EQ.3)GO TO 136
C   CHARGE DISTRIBUTION TYPE 1
      WRITE(I00OUT,831)
831 FORMAT(/,31X,42HSIMULTANEOUS BURST , UNIFORMLY DISTRIBUTED,
+27H CHARGES IN A PARALLELOGRAM)
      NCH=NCHS(1)*NCHS(2)
      WRITE(I00OUT,832)NCH,(SRCBAS(I),I=1,2)
832 FORMAT(27X,28HTOTAL NUMBER OF CHARGES IS ,1X,I3,1X,
+27H WITH REFERENCE CHARGE AT ,(F8.2,1H,,F8.2,1H))
      WRITE(I00OUT,834)NCHS(1),(SIDE1(I),I=1,2)
834 FORMAT(32X,I3,1X,45HCHARGES WITH DIRECTION AND SPACING GIVEN BY ,
+F8.2,1H,,F8.2,1H>)
      WRITE(I00OUT,834)NCHS(2),(SIDE2(I),I=1,2)
      GO TO 300
C   CHARGE DISTRIBUTION TYPE 2
133 NCH=NCHS(1)
      DO 134 J=1,NCH
      READ(I0IN,701)(COOR(K,J),K=1,2)
701 FORMAT(8X,2F8.2)
134 CONTINUE
      WRITE(I00OUT,836)
836 FORMAT(/,42X,48HSIMULTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES)
      WRITE(I00OUT,838)NCHS(1)
838 FORMAT(51X,26HTOTAL NUMBER OF CHARGES IS ,1X,I3 )
      WRITE(I00OUT,840)
840 FORMAT(55X,22HDETONATION COORDINATES)
      DO 135 J=1,NCH
      WRITE(I00OUT,842)(COOR(I,J),I=1,2)
842 FORMAT(53X,2(3X,F8.2))
135 CONTINUE
      GO TO 300
C   CHARGE DISTRIBUTION TYPE 3
136 NCH=NCHS(1)
      DO 137 J=1,NCH
      READ(I0IN,702)(COOR(K,J),K=1,2),TSTAG(J)
702 FORMAT(8X,3F8.2)
137 CONTINUE
      WRITE(I00OUT,844)
844 FORMAT(/,30X,38HSEQUENTIAL IN TIME AND RANDOM IN SPACE,
+24H DISTRIBUTION OF CHARGES)
      WRITE(I00OUT,838)NCH
      WRITE(I00OUT,846)
846 FORMAT(45X,25H DETONATION COORDINATES ,7X,10HBLAST TIME)
      DO 138 J=1,NCH
      WRITE(I00OUT,848)(COOR(I,J),I=1,2),TSTAG(J)
848 FORMAT(46X,F8.2,3X,F8.2,12X,F8.2)
138 CONTINUE
      GO TO 300
C   STORE AND PRINT OUT INFORMATION ABOUT VEHICLE
150 CONTINUE
      VH=.TRUE.

```

```

V0(1)=RDIN(2) DRT06200
V0(2)=RDIN(3) DRT06210
VEHDIR=RDIN(4) DRT06220
VEHSPO=RDIN(5) DRT06230
VEHWID=RDIN(6) DRT06240
VEHWHT=RDIN(7) DRT06250
VEHTYP=IFIX(RDIN(8)) DRT06260
850 WRITE(I00UT,850)V0(1),V0(2) DRT06270
FORMAT(//,44X,26HINITIAL VEHICLE POSITION ,F8.2,1H,,F8.2,1H) DRT06280
852 FORMAT(44X,19HVEHICLE DIRECTION ,F8.2,17H (CCW FROM EAST)) DRT06290
854 FORMAT(50X,20HVEHICLE SPEED ,F8.2,4H M/S) DRT06300
856 FORMAT(52X,20HVEHICLE WIDTH ,F8.2) DRT06310
858 FORMAT(52X,20HVEHICLE WEIGHT ,F8.2) DRT06320
859 IF(VEHTYP.EQ.0)WRITE(I00UT,891) DRT06330
891 FORMAT(58X,15HWHEELED VEHICLE) DRT06340
892 IF(VEHTYP.EQ.1)WRITE(I00UT,892) DRT06350
892 FORMAT(58X,15HTRACKED VEHICLE) DRT06360
GO TO 300 DRT06370
C STORE TRANSMITTER COORDINATES AND TRANSMISSION THRESHOLD DRT06380
C
170 CONTINUE DRT06390
TC=.TRUE. DRT06400
NEWCOR=.TRUE. DRT06410
TRNCOR(1)=RDIN(2) DRT06420
TRNCOR(2)=RDIN(3) DRT06430
TRNCOR(3)=RDIN(4) DRT06440
TRNMIN=RDIN(5) DRT06450
IF(TRNMIN.LT.1.E-05)TRNMIN=1.E-05 DRT06460
THRESH=-ALOG(TRNMIN) DRT06470
GO TO 300 DRT06480
C STORE RECEIVER COORDINATES DRT06490
C
190 CONTINUE DRT06500
RC=.TRUE. DRT06510
RECCOR(1)=RDIN(2) DRT06520
RECCOR(2)=RDIN(3) DRT06530
RECCOR(3)=RDIN(4) DRT06540
GO TO 300 DRT06550
C STORE OBSERVER COORDINATES DRT06560
C
210 CONTINUE DRT06570
OC=.TRUE. DRT06580
OBSCOR(1)=RDIN(2) DRT06590
OBSCOR(2)=RDIN(3) DRT06600
SPCHT=RDIN(4) DRT06610
GO TO 300 DRT06620
C STORE TIME INTERVAL FOR CALCULATIONS DRT06630
C
230 CONTINUE DRT06640
TM=.TRUE. DRT06650
TSTART=RDIN(2) DRT06660
TEND=RDIN(3) DRT06670
TINC=RDIN(4) DRT06680
IF(TINC.LE.0.0)TINC=1. DRT06690
IF(TEND.LT.TSTART)GO TO 903 DRT06700
LIM=IFIX((TEND-TSTART)/TINC)+1 DRT06710
300 CONTINUE DRT06720
310 CONTINUE DRT06730
IF(IGEOSW.NE.1) GO TO 333 DRT06740
TRNCOR(1)=PTS(1)*1000. DRT06750
TRNCOR(2)=PTS(2)*1000. DRT06760
TRNCOR(3)=PTS(3)*1000. DRT06770
DRT06780
DRT06790
DRT06800
DRT06810
DRT06820
DRT06830
DRT06840
DRT06850
DRT06860
DRT06870
DRT06880

```

```

RECCOR(1)=PTS(4)*1000. DRT06890
RECCOR(2)=PTS(5)*1000. DRT06900
RECCOR(3)=PTS(6)*1000. DRT06910
OBSCOR(1)=PTS(10)*1000. DRT06920
OBSCOR(2)=PTS(11)*1000. DRT06930
SPCHT=PTS(12)*1000. DRT06940
333 CONTINUE DRT06950
IOPT=IFIX(RDIN(2)) DRT06970
IFILE=IFIX(RDIN(3)) DRT06980
IF(IOPT.NE.3) GO TO 305 DRT06990
WRITE(IOUT,301) DRT07000
WRITE(IOUT,302) DRT07010
WRITE(IOUT,303) IFILE DRT07020
WRITE(IOUT,304) DRT07030
WRITE(IOUT,305) DRT07040
301 FORMAT(1H0,130(1H*),/) DRT07050
302 FORMAT(1H0,58X,15HDIRTRAN WARNING,/) DRT07060
303 FORMAT(1H0,38X,13HLOGICAL UNIT,I2,27H IS ASSIGNED TO A TEMPORARY,DRT07070
+ 13H STORAGE FILE) DRT07080
304 FORMAT(1H0,28X,46HCARE MUST BE TAKEN TO INSURE THAT THIS UNIT IS, DRT07090
+27H NOT IN USE BY ANOTHER FILE,/) DRT07100
305 CONTINUE DRT07110
IF((IOPT.EQ.1.OR.IOPT.EQ.2).AND.(EX.AND.(&.NOT.CH)))GO TO 909 DRT07120
IF((IOPT.EQ.1.OR.IOPT.EQ.2).AND.(&CH.AND.(&.NOT.EX)))GO TO 909 DRT07130
IF((IOPT.EQ.3).AND.OC)OC=.FALSE. DRT07140
IF((IOPT.EQ.4).AND.OC)OC=.FALSE. DRT07150
IF((IOPT.EQ.5).AND.OC)OC=.FALSE. DRT07160
IF((TC.AND.(&.NOT.RC)).OR.(&RC.AND.(&.NOT.TC)))GO TO 913 DRT07170
IF(M1,.OR.M2)NEWATM=.TRUE. DRT07180
IF(IOPT.EQ.3.AND.CH)NEWSRC=.TRUE. DRT07190
IF(EX.AND.CH)NEWSRC=.TRUE. DRT07200
IF(TC.AND.RC)LOSTRN=.TRUE. DRT07210
IF(VH)EDGE=.TRUE. DRT07220
IF(VH)NEWVEH=.TRUE. DRT07230
IF(IOPT.EQ.3)LIM=1 DRT07240
DO 400 J=1,LIM DRT07250
TIME=TSTART+TINC*FLOAT(J-1) DRT07260
NEWTIM=.TRUE. DRT07270
NERR=0 DRT07280
IF(IOPT.EQ.1.OR.IOPT.EQ.2)GO TO 320 DRT07290
IF(IOPT.EQ.3.OR.IOPT.EQ.4)GO TO 325 DRT07300
IF((IOPT.EQ.1).AND.(NARY.NE.1))GO TO 915 DRT07310
IF((IOPT.EQ.2).AND.(NARY.NE.2))GO TO 915 DRT07320
IF((IOPT.EQ.4).AND.(NARY.NE.3))GO TO 915 DRT07330
C COMPUTE FOR VEHICLE SOURCE DRT07340
C CHECK TO SEE IF WE HAVE THE MINIMUM INPUT REQUIREMENTS DRT07350
C
IF(DSOD.GT.0.0)GO TO 315 DRT07360
IF(.NOT.(M1.AND.M2.AND.SL.AND.VH.AND.TC.AND.RC.AND.TM))GO TO 911 DRT07370
CALL VEHCL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,PHI,NSOIL, DRT07380
1 SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR, DRT07390
2 VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH,TRNL0S,NERR) DRT07400
1 NEWVEH=.FALSE. DRT07410
2 NEWATM=.FALSE. DRT07420
GO TO 330 DRT07430
315 TRNL0S=1.0 DRT07440
GO TO 335 DRT07450
325 CONTINUE DRT07460
IF(IOPT.EQ.4.AND.(&.NOT.(EX.AND.TC.AND.RC.AND.TM)))GO TO 911 DRT07470
IF(IOPT.EQ.3.AND.(&.NOT.(M1.AND.M2.AND.SL.AND.CH)))GO TO 911 DRT07480
CALL COMPCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND, DRT07490
1 PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,NWL, DRT07500
2 TRNCOR,RECCOR,TIME,DHDT,TRNL0S,NERR) DRT07510
1 NEWSRC=.FALSE. DRT07520
2 NEWATM=.FALSE. DRT07530
IF(IOPT.EQ.3)GO TO 410 DRT07540
GO TO 330 DRT07550
320 CONTINUE DRT07560
DRT07570
DRT07580
DRT07590

```

```

C C CHECK TO SEE IF MINIMUM INPUTS ARE AVAILABLE DRT07600
C C IF(.NOT.(M1.AND.M2.AND.SL.AND.CH.AND.EX.AND.(TC.AND.RC.OR.OC).AND.DRT07610
1 T1M))GO TO 911 DRT07620
1 CALL DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWHD,WNDMES,PHI, DRT07630
1 THWHD,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD, DRT07640
2 LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM, DRT07650
3 TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS, DRT07660
4 NERR,NCHS,SRCBAS,SIDE1,SIDE2,DHDT) DRT07670
NEWSPC=.FALSE. DRT07680
NEWATM=.FALSE. DRT07690
DRT07700
330 IF(NERR.EQ.0)GO TO 335 DRT07710
335 WRITE(I00UT,857)NERR DRT07720
857 FORMAT(55X,30H ***** DIRTRAN ERROR NUMBER ,I2) DRT07730
GO TO 400 DRT07740
335 IF(ONCE.AND.(&.NOT.NEWCOR))GO TO 340 DRT07750
NEWCOR=.FALSE. DRT07760
IZINV=IFIX(ZINV) DRT07770
WRITE(I00UT,859)IZINV DRT07780
859 FORMAT(//,47X,30HESTIMATED INVERSION HEIGHT ,I7) DRT07790
C C IF OBSERVER IS SPECIFIED, OUTPUT IS LABELED FOR EACH TIME. DRT07800
C C IF ONLY TRANSMITTER AND RECEIVER ARE INPUT, OUTPUT IS TABULAR DRT07810
C C IF(TC.AND.RC.AND.OC)GO TO 350 DRT07820
IF(OI)GO TO 350 DRT07830
IF(.NOT.(TC.AND.RC))GO TO 905 DRT07840
DRT07850
WRITE(I00UT,860)WAVE1 DRT07860
860 FORMAT(//,47X,18HWAVELENGTH ,F7.2,12H MICROMETERS) DRT07870
DRT07880
WRITE(I00UT,862)(TRNCOR(I),I=1,3) DRT07890
DRT07900
862 FORMAT(37X,28HTRANSMITTER COORDINATES ,3F10.2) DRT07910
864 FORMAT(37X,28HRECEIVER COORDINATES ,3F10.2) DRT07920
DRT07930
WRITE(I00UT,866) DRT07940
866 FORMAT(52X,4HTIME,10X,13HTRANSMITTANCE) DRT07950
340 CONTINUE DRT07960
340 WRITE(I00UT,868)TIME,TRNLOS DRT07970
868 FORMAT(52X,F8.2,10X,E10.5) DRT07980
ONCE=.TRUE. DRT07990
GO TO 400 DRT08000
350 WRITE(I00UT,923)TIME DRT08010
923 FORMAT(//,48X,28HTIME AFTER BLAST ,F7.2) DRT08020
IF(.NOT.(TC.AND.RC))GO TO 360 DRT08030
WRITE(I00UT,808) DRT08040
DRT08050
WRITE(I00UT,860)WAVE1 DRT08060
DRT08070
WRITE(I00UT,862)(TRNCOR(I),I=1,3) DRT08080
DRT08090
WRITE(I00UT,864)(RECCOR(I),I=1,3) DRT08100
DRT08110
870 FORMAT(42X,38HTRANSMITTANCE ALONG THE LINE OF SIGHT ,E10.3) DRT08120
360 WRITE(I00UT,808) DRT08130
DRT08140
WRITE(I00UT,872) DRT08150
DRT08160
972 FORMAT(57X,26HAERODYNAMIC CLOUD DIMENSIONS) DRT08170
DRT08180
WRITE(I00UT,808) DRT08190
DRT08200
WRITE(I00UT,874)(OBSCOR(I),I=1,2) DRT08210
DRT08220
874 FORMAT(41X,28HOBSERVER COORDINATES ,2F10.2) DRT08230
DRT08240
WRITE(I00UT,876)HEIGHT DRT08250
DRT08260
876 FORMAT(39X,26HTHE HEIGHT OF THE CLOUD IS,10X,F10.2,7H METERS) DRT08270
DRT08280
878 FORMAT(38X,28HTHE CENTROID COORDINATES ARE,8X,2F10.2) DRT08290
DRT08300
WRITE(I00UT,880)CENWTH DRT08310
DRT08320
880 FORMAT(38X,28HTHE WIDTH AT THE CENTROID IS,8X,F10.2, 7H METERS) DRT08330
DRT08340
WRITE(I00UT,882)SPCHT,SPCWTH DRT08350
DRT08360
882 FORMAT(39X,12HTHE WIDTH AT,F8.2,11H METERS IS ,5X,F10.2,7H METERS) DRT08370
DRT08380
WRITE(I00UT,884)NCPTS DRT08390
DRT08400
884 FORMAT(46X,13,37H CONTOUR POINTS HAVE BEEN DETERMINED ) DRT08410
DRT08420
WRITE(I00UT,886)((CPTSC(I0,IPT),I0=1,2),IPT=1,NCPTS) DRT08430
DRT08440
886 FORMAT(60X,2(F10.3,2X))) DRT08450
DRT08460
400 CONTINUE DRT08470
GO TO 10 DRT08480
DRT08490

```

```

410 WRITE(I00UT,888) DRT08300
888 FORMAT(/,38X,48H*** THE CLOUD HAS BEEN PRECOMPUTED AND STORED ON,
1 5H FILE) DRT08310
GO TO 10 DRT08320
900 WRITE(I00UT,901) DRT08330
901 FORMAT(/,24X,48H*** DIRTRAN ERROR - MORE THAN 15 RECORDS OF DATA,
1 35H HAVE BEEN INPUT WITHOUT A GO CARD.) DRT08340
IERR=1 DRT08350
GO TO 999 DRT08360
903 WRITE(I00UT,904) DRT08370
904 FORMAT(/,39X,47H*** DIRTRAN ERROR - TIMES ARE NOT IN INCREASING,
+ 6H ORDER) DRT08380
IERR=1 DRT08390
GO TO 999 DRT08400
905 WRITE(I00UT,906) DRT08410
906 FORMAT(/18X,46H*** DIRTRAN ERROR -NO TRANSMITTER AND RECEIVER,
+ 49H AND/OR OBSERVER COORDINATES HAVE BEEN SPECIFIED.) DRT08420
IERR=1 DRT08430
GO TO 999 DRT08440
909 WRITE(I00UT,910) DRT08450
910 FORMAT(/,25X,44H*** DIRTRAN ERROR - ONLY ONE DATA RECORD FOR,
1 38H CHARGE INFORMATION HAS BEEN SPECIFIED) DRT08460
IERR=1 DRT08470
GO TO 999 DRT08480
911 WRITE(I00UT,912) DRT08490
912 FORMAT(/,16X,49H*** DIRTRAN ERROR - MINIMUM AMOUNT OF INFORMATION,
1 26H REQUIRED IS NOT AVAILABLE,/,,10X,14H CHECK INPUTS) DRT08500
IERR=1 DRT08510
GO TO 999 DRT08520
913 WRITE(I00UT,914) DRT08530
914 FORMAT(/20X,49H*** DIRTRAN ERROR - BOTH TRANSMITTER AND RECEIVER,
1 43H LOCATIONS MUST BE SPECIFIED, CHECK INPUTS) DRT08540
IERR=1 DRT08550
GO TO 999 DRT08560
915 WRITE(I00UT,916) DRT08570
916 FORMAT(/30X,50H IOPT AND NARY DO NOT AGREE SEE THE ABOVE COMMENTS,
1 21H FOR CORRECT MATCHING) DRT08580
IERR=1 DRT08590
999 RETURN DRT08600
END DRT08610
DRT08620
DRT08630
DRT08640
DRT08650
DRT08660
DRT08670
DRT08680
DRT08690

```

SUBROUTINE AMOUNT(VOLSPH,WAKAL,SPHAL)	AMOU0250
SUBROUTINE TO DETERMINE LOADING FOR THE SPHERE AND WAKE	AMOU0010
INPUTS	AMOU0020
VOLSPH - VOLUME OF THE BUOYANTLY RISING SPHERE	AMOU0030
ALL OTHER NEEDED INFORMATION IS PASSED VIA COMMON BLOCKS	AMOU0040
OUTPUTS	AMOU0050
WAKAL - AMOUNT OF INITIAL LOADING OF SPHERE THAT HAS BEEN	AMOU0060
DEPOSITED IN THE WAKE	AMOU0070
SPHAL - AMOUNT OF INITIAL LOADING OF SPHERE THAT IS LEFT	AMOU0080
IN THE SPHERE	AMOU0090
FUNCTIONS AND SUBROUTINES NEEDED	AMOU0100
NONE	AMOU0110
*****	AMOU0120
COMMON/NTAL/TNOT,VOLNOT,TNO,CBLEED	AMOU0130
COMMON/BUDYCL/RSPH,DELT,Y2,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM	AMOU0140
TSPH=TNO+DELT	AMOU0150
STUFF=CBLEED*(VOLSPH/TSPH-VOLNOT/TNOT)	AMOU0160
WAKAL=AMINI(SPHNS(1),STUFF)	AMOU0170
SPHAL=SPHNS(1)-WAKAL	AMOU0180
RETURN	AMOU0190
END	AMOU0200
	AMOU0210
	AMOU0220
	AMOU0230
	AMOU0240
	AMOU0250
	AMOU0260
	AMOU0270
	AMOU0280
	AMOU0290
	AMOU0300
	AMOU0310
	AMOU0320
	AMOU0330

```

SUBROUTINE ATMCAL(NATM,ZT,TMES,ZU,UMES,PHI,BETA,DHDT,ERR)          ATMC0010
REAL M,N,K,KM                                                 ATMC0020
LOGICAL ERR,DHDT                                              ATMC0030
DIMENSION ZT(2),TMES(2),ZU(2),UMES(2),ZL0(6)                      ATMC0040
COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,M,N,ZINV                      ATMC0050
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                           ATMC0060
COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT                         ATMC0070
COMMON/STAR5/USTAR,TSTAR,ZSTAR                                ATMC0080
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUATMC0090
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK                  ATMC0100
DATA ZL0/-2.5,-4.5,-13.5,10000.,55.,20./                      ATMC0110
DATA OMEGA,K/7.2722E-05,.4/                                     ATMC0120
*****                                                       ATMC0130
ATMC0140
ATMC0150
ATMC0160
ATMC0170
ATMC0180
ATMC0190
ATMC0200
ATMC0210
ATMC0220
ATMC0230
ATMC0240
ATMC0250
ATMC0260
ATMC0270
ATMC0280
ATMC0290
ATMC0300
ATMC0310
ATMC0320
ATMC0330
ATMC0340
ATMC0350
ATMC0360
ATMC0370
ATMC0380
ATMC0390
ATMC0400
ATMC0410
ATMC0420
ATMC0430
ATMC0440
ATMC0450
ATMC0460
ATMC0470
ATMC0480
ATMC0490
ATMC0500
ATMC0510
ATMC0520
ATMC0530
ATMC0540
ATMC0550
ATMC0560
ATMC0570
ATMC0580
ATMC0590
ATMC0600
ATMC0610
ATMC0620
ATMC0630
ATMC0640
ATMC0650
ATMC0660
ATMC0670
ATMC0680
ATMC0690
ATMC0700

```

C ****

PURPOSE

TO FIT THE BEST POWER-LAW PROFILES OF WINDSPEED AND DIFFUSIVITY CONSISTENT WITH KNOWN RELATIONS GOVERNING THE CONSTANT SHEAR STRESS LAYER TO GIVEN MEASUREMENTS AT ONE OR TWO HEIGHTS. ALSO TO CALCULATE PARAMETERS NEEDED FOR VERTICAL VARIATION IN WIND DIRECTION LAYER, FOR WIND AND TEMPERATURE PROFILES.

INPUTS

NATM	INTEGER WHICH IS 0 IF WINDSPEED AND TEMPERATURE ARE AVAILABLE AT TWO HEIGHTS AND EQUAL TO THE PASQUILL CATEGORY OTHERWISE.	ATMC0240
ZT	SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS (IN METERS) AT WHICH TEMPERATURES WILL BE GIVEN. MUST BE IN ASCENDING ORDER.	ATMC0250
TMES	SINGLY DIMENSIONED ARRAY CONTAINING THE TWO TEMPERATURE MEASUREMENTS IN DEGREES KELVIN AT HEIGHTS ZT.	ATMC0260
ZU	SINGLY DIMENSIONED ARRAY CONTAINING ONE OR TWO HEIGHTS (METERS) AT WHICH WIND SPEEDS WILL BE GIVEN. MUST BE IN ASCENDING ORDER.	ATMC0270
UMES	SINGLY DIMENSIONED ARRAY CONTAINING THE ONE OR TWO WIND SPEED MEASUREMENTS (M/S) AT HEIGHTS UMES.	ATMC0280
PHI	LATITUDE OF DETONATION SITE.	ATMC0290
BETA	ANGLE OF WIND VELOCITY VECTOR MEASURED COUNTER-CLOCKWISE FROM EAST.	ATMC0300
DHDT	A LOGICAL VARIABLE WHICH IS .FALSE. IF THE INVERSION LAYER HEIGHT IS RELATIVELY CONSTANT AND .TRUE. IF THE LAYER HEIGHT IS INCREASING.	ATMC0310

OUTPUTS

ERR	A LOGICAL WHICH IS TRUE IF AN ERROR IS INCURRED DURING THE CALCULATION.	ATMC0320
DXZ0	THE RATIO OF THE DIFFUSIVITY IN THE X DIRECTION TO THE DIFFUSIVITY IN THE Z DIRECTION. RETURNED IN COMMON /WNDPRM/.	ATMC0330
DYX0	THE RATIO OF THE DIFFUSIVITY IN THE Y DIRECTION TO THE DIFFUSIVITY IN THE X DIRECTION. RETURNED IN COMMON /WNDPRM/.	ATMC0340

D20	THE COEFFICIENT OF Z**N IN THE VERTICAL PROFILE OF VERTICAL DIFFUSIVITY. RETURNED IN COMMON /WNDPRM/.	ATMC0710 ATMC0720 ATMC0730 ATMC0740 ATMC0750 ATMC0760 ATMC0770 ATMC0780
U0	THE COEFFICIENT OF Z**M IN THE VERTICAL PROFILE OF HORIZONTAL WIND SPEED. RETURNED IN COMMON /WNDPRM/.	ATMC0790 ATMC0800 ATMC0810 ATMC0820 ATMC0830 ATMC0840
M	THE EXPONENT OF Z IN THE HORIZONTAL WIND SPEED PROFILE. RETURNED IN COMMON /WNDPRM/.	ATMC0850 ATMC0860 ATMC0870 ATMC0880 ATMC0890 ATMC0900 ATMC0910 ATMC0920
N	THE EXPONENT OF Z IN THE VERTICAL DIFFUSIVITY PROFILE. RETURNED IN COMMON /WNDPRM/.	ATMC0930 ATMC0940 ATMC0950 ATMC0960
ZINV	ESTIMATED INVERSION HEIGHT. RETURNED IN /WNDPRM/.	ATMC0970 ATMC0980
USTAR	VELOCITY PROFILE SCALE RETURNED IN COMMON /STARS/.	ATMC0990 ATMC1000
TSTAR	TEMPERATURE PROFILE SCALE. RETURNED IN COMMON /STARS/.	ATMC1010 ATMC1020 ATMC1030
ZSTAR	HEIGHT AT WHICH THE VERTICAL VARIATION IN WIND DIRECTION PROFILES FOR WIND AND TEMPERATURE TAKE EFFECT.	ATMC1040 ATMC1050 ATMC1060
	OUTPUT RETURNED IN COMMON /EKWIND/ ARE THE PARAMETERS NEEDED FOR THE WIND PROFILE ABOVE ZSTAR.	ATMC1070 ATMC1080
	OUTPUT RETURNED IN COMMON /EKTEMP/ ARE THE PARAMETERS NEEDED FOR THE TEMPERATURE PROFILES ABOVE ZSTAR.	ATMC1090 ATMC1100 ATMC1110 ATMC1120 ATMC1130 ATMC1140 ATMC1150 ATMC1160 ATMC1170
CALLED FROM DUSTCL		
NEEDED FUNCTIONS AND SUBROUTINES		
TMPCAL	CALCULATES SCALED TEMPERATURE PROFILES	ATMC1180 ATMC1190
WNDCAL	CALCULATES SCALED WIND SPEED PROFILES	ATMC1200 ATMC1210
DIFFUS	FUNCTION TO CALCULATE THE DIFFUSIVITY AT A GIVEN HEIGHT.	ATMC1220 ATMC1230 ATMC1240
TEMP	CALCULATES THE POTENTIAL TEMPERATURE AND GRADIENT AT A GIVEN HEIGHT.	ATMC1250 ATMC1260 ATMC1270 ATMC1280 ATMC1290 ATMC1300 ATMC1310 ATMC1320 ATMC1330 ATMC1340 ATMC1350 ATMC1360

ERR=.FALSE.		
DELTH IS THE DIFFERENCE IN POTENTIAL TEMPERATURE BETWEEN THE TWO HEIGHTS WHERE TEMPERATURE IS GIVEN.		
Z0=0.01 T0=TMES(1) IF(NATM.EQ.0)GO TO 100		ATMC1370 ATMC1380 ATMC1390 ATMC1400
ASSIGN ATMOSPHERIC PROFILE ACCORDING TO SPECIFIED PASQUILL CATEGORY		
ZL	FRICTION HEIGHT	ATMC1320
ZL	MONIN-OBUKOV LENGTH	ATMC1330
USTAR	THE FRICTION VELOCITY	ATMC1340
TSTAR	THE SCALING TEMPERATURE	ATMC1350
ZL=ZL0(NATM) IF(NATM.GE.5)Z0=1.E-04*ABS(ZL) IF(NATM.LE.3)Z0=1.E-03*ABS(ZL) NP=IFIX(SIGN(1.,ZL))		ATMC1360 ATMC1370 ATMC1380 ATMC1390

```

USTAR=UMES(1)/WNDCAL(Z0,ZL,ZU(1))
TSTAR=TMES(1)*USTAR**2/1.568/ZL
IF(NATM-4)200,300,210
100 CONTINUE
C USE ITERATIVE PROCEDURE TO CONVERGE ON BEST ATMOSPHERIC PROFILE
C TO MATCH DATA AT TWO HEIGHTS
C
DELTH=TMES(2)-TMES(1)+.0098*(ZT(2)-ZT(1))
NP=SIGN(1,DELTH)
DELU=UMES(2)-UMES(1)
ZULOG=ALOG(ZU(2)/ZU(1))
ZTLOG=ALOG(ZT(2)/ZT(1))
USTAR=(UMES(2)-UMES(1))/ZULOG
TSTAR=DELTH/ZTLOG
ZL=.638*TMES(1)*USTAR**2/TSTAR
IF(ABS(ZL).GE.1.000,GO TO 300
DO 110 ITER=1,100
USTAR=DELU/(WNDCAL(Z0,ZL,ZU(2))-WNDCAL(Z0,ZL,ZU(1)))
TSTAR=DELTH/(TMPCAL(Z0,ZL,ZT(2))-TMPCAL(Z0,ZL,ZT(1)))
ZLP=ZL
ZL=.638*TMES(1)*USTAR**2/TSTAR
IF(ABS((ZL-ZLP)/ZLP).LT..01)GO TO 120
110 CONTINUE
ERR=.TRUE.
GO TO 999
120 CONTINUE
IF(ZL.GT.0.)Z0=1.E-04*ABS(ZL)
IF(ZL.LE.0.)Z0=1.E-03*ABS(ZL)
IF(NP)200,300,210
200 CONTINUE
C UNSTABLE ATMOSPHERE
C
DXZ0=2.6
M=.079943
N=4./3.
DZ0=.7609*USTAR*ABS(ZL)**(1.-N)
U0=USTAR+14.2478/ABS(ZL)**M
GO TO 430
210 CONTINUE
C STABLE ATMOSPHERE
C
DXZ0=3.3
N=.45644
M=.28414
DZ0=.059517*USTAR*ABS(ZL)**(1.-N)
U0=USTAR+36.6642/ABS(ZL)**M
GO TO 430
300 CONTINUE
C NEUTRAL ATMOSPHERE
C
DZ0=.4*USTAR
DXZ0=2.8
NP=0
N=1.
M=1./7.
U0=45.92*USTAR/ABS(ZL)**M
430 CONTINUE
C COMMON CALCULATION TO UNSTABLE, NEUTRAL, AND STABLE ATMOSPHERES
C
DYX0=1.
IF(NATM.EQ.0)U0=(U0+UMES(2)/ZU(2)**M)/2.
C ESTIMATE THE INVERSION HEIGHT AND COMPUTE THE NECESSARY
C PARAMETERS FOR THE WIND AND TEMPERATURE PROFILES BETWEEN
C ZSTAR AND ZINY WHEN DHDT IS .FALSE..
ATMC1410
ATMC1420
ATMC1430
ATMC1440
ATMC1450
ATMC1460
ATMC1470
ATMC1480
ATMC1490
ATMC1500
ATMC1510
ATMC1520
ATMC1530
ATMC1540
ATMC1550
ATMC1560
ATMC1570
ATMC1580
ATMC1590
ATMC1600
ATMC1610
ATMC1620
ATMC1630
ATMC1640
ATMC1650
ATMC1660
ATMC1670
ATMC1680
ATMC1690
ATMC1700
ATMC1710
ATMC1720
ATMC1730
ATMC1740
ATMC1750
ATMC1760
ATMC1770
ATMC1780
ATMC1790
ATMC1800
ATMC1810
ATMC1820
ATMC1830
ATMC1840
ATMC1850
ATMC1860
ATMC1870
ATMC1880
ATMC1890
ATMC1900
ATMC1910
ATMC1920
ATMC1930
ATMC1940
ATMC1950
ATMC1960
ATMC1970
ATMC1980
ATMC1990
ATMC2000
ATMC2010
ATMC2020
ATMC2030
ATMC2040
ATMC2050
ATMC2060
ATMC2070
ATMC2080
ATMC2090
ATMC2100

```

ATMC2110
 ATMC2120
 ATMC2130
 ATMC2140
 ATMC2150
 ATMC2160
 ATMC2170
 ATMC2180
 ATMC2190
 ATMC2200
 ATMC2210
 ATMC2220
 ATMC2230
 ATMC2240
 ATMC2250
 ATMC2260
 ATMC2270
 ATMC2280
 ATMC2290
 ATMC2300
 ATMC2310
 ATMC2320
 ATMC2330
 ATMC2340
 ATMC2350
 ATMC2360
 ATMC2370
 ATMC2380
 ATMC2390
 ATMC2400
 ATMC2410
 ATMC2420

```

C
APHI=PHI*PIRAD
FREQ=2.*OMEGA*SINK(APHI)
HC=K*USTAR/FREQ
IF(ZL.GT.0.0.AND.ZL.LT.1.E3)GO TO 500
ZINV=HC
GO TO 501
500 ZINV=.26*HC
501 ZSTAR=.13*ZINV
KM=DIFFUS(Z0,ZL,ZSTAR)
SP=USTAR*WNDCAL(Z0,ZL,ZSTAR)
ALP=SQRT(FREQ/(2.*KM))
IF(DHDT)GO TO 813
ARG=ALP*ZSTAR
ARG1=BETA+ALP*ZINV
ARG2=ALP*(ZINV-ZSTAR)
C=SP*EXP(-ARG)*SINK(ARG1)/SIN(ARG2)
PYF=C*EXP(-ARG)*COS(ARG)-SP*COS(BETA)
PXF=C*EXP(-ARG)*SINK(ARG)+SP*SINK(BETA)
UE=C*EXP(-ARG)*COS(ARG)-PYF
VE=-C*EXP(-ARG)*SINK(ARG)+PXF
UHAT=UE/SQRT(UE*UE+VE*VE)
VHAT=VE/SQRT(UE*UE+VE*VE)
CALL TEMP(ZSTAR,TA,DTADZ)
DTADH=0.0
TC3=(DTADH-DTADZ)/(2.*(ZINV-ZSTAR))
TC2=DTADZ-2.*TC3*ZSTAR
TC1=TA-TC2*ZSTAR-TC3*ZSTAR**2
GO TO 999
813 ZSTAR=1.E4
999 RETURN
END

```

```

SUBROUTINE AVRG(ZX,TIME,QTOT,XBAVG,SIG2X,SIG2Y)          AVRG0230
ROUTINE FOR FINDING AVERAGES OF THE MOMENTS OF THE DISCS   AVRG0010
AVRG0020
AVRG0030
AVRG0040
AVRG0050
AVRG0060
AVRG0070
AVRG0080
AVRG0090
AVRG0100
AVRG0110
AVRG0120
AVRG0130
AVRG0140
AVRG0150
AVRG0160
AVRG0170
AVRG0180
AVRG0190
AVRG0200
AVRG0210
AVRG0220
COMMON/DISCS/NDSQS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3) AVRG0240
COMMON/PRTINF/R0,YGRAV(3),NPRTS                         AVRG0250
QTOT=0.0                                                    AVRG0260
QSIG2X=0.0                                                   AVRG0270
QSIG2Y=0.0                                                   AVRG0280
QXBAR=0.0                                                   AVRG0290
Z=ZX                                                       AVRG0300
DO 10 I=1,NDSQS                                         AVRG0310
H=ZDSC(I)                                                 AVRG0320
ROH2=R2DSC(I)                                             AVRG0330
TOF=TIME-TDSC(I)                                         AVRG0340
CALL MOMENT(YGRAV(1),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)        AVRG0350
QT=QDSC(I,1)*Q                                           AVRG0360
QTOT=QTOT+QT                                             AVRG0370
QSIG2X=QSIG2X+(SIGW2+ROH2/2.)*QT                         AVRG0380
QSIG2Y=QSIG2Y+(SIGP2+ROH2/2.)*QT                         AVRG0390
QXBAR=QXBAR+(XBAR+XDSC(I))*QT                           AVRG0400
10 CONTINUE                                                AVRG0410
XBAVG=QXBAR/QTOT                                         AVRG0420
SIG2X=QSIG2X/QTOT                                         AVRG0430
SIG2Y=QSIG2Y/QTOT                                         AVRG0440
999 RETURN                                                 AVRG0450
END                                                       AVRG0460

```

SUBROUTINE CLIMB(FUNCT,GFUN,P1,FP1,NSEARCH,NOCONT) CLI00520
 THIS MODULE IS A SUBROUTINE THAT FINDS A POINT ON A CONTOUR CLI00010
 BY FINDING THE GRADIENT VECTOR AT THAT POINT AND MARCHING ALONG CLI00020
 IT UNTIL IT FINDS ITSELF IN A REGION GREATER THAN THE CONTOUR LEVEL. CLI00030
 AT WHICH POINT IT MARCHES HORIZONTALLY, HALVING THE STEP SIZE CLI00040
 UNTIL THE CONTOUR IS REACHED WITHIN SPECIFIED RESOLUTION. CLI00050
 IN ADDITION IT WILL DETERMINE IF A CONTOUR EXISTS. CLI00060
 CLI00070
 CLI00080
 CLI00090
 CLI00100
 CLI00110
 CLI00120
 CLI00130
 CLI00140
 CLI00150
 CLI00160
 CLI00170
 CLI00180
 CLI00190
 CLI00200
 CLI00210
 CLI00220
 CLI00230
 CLI00240
 CLI00250
 CLI00260
 CLI00270
 CLI00280
 CLI00290
 CLI00300
 CLI00310
 CLI00320
 CLI00330
 CLI00340
 CLI00350
 CLI00360
 CLI00370
 CLI00380
 CLI00390
 CLI00400
 CLI00410
 CLI00420
 CLI00430
 CLI00440
 CLI00450
 CLI00460
 CLI00470
 CLI00480
 CLI00490
 CLI00500
 CLI00510
 CLI00520
 CLI00530
 CLI00540
 CLI00550
 CLI00560
 CLI00570
 CLI00580
 CLI00590
 CLI00600
 CLI00610
 CLI00620
 CLI00630
 CLI00640
 CLI00650
 CLI00660
 CLI00670
 CLI00680
 CLI00690
 CLI00700

ARGUMENTS PASSED.
 INPUT
 FUNCT-THE FUNCTION(X,Y) ALSO GIVEN IN EXTERNAL.
 P1-THE STARTING POINT.

OUTPUT
 P1 - THE POINT ON THE CONTOUR OR THE POINT AT WHICH
 THE FUNCTION REACHES A MAXIMUM BELOW THE CONTOUR
 LEVEL CLI00200
 FP1 - THE VALUE OF THE FUNCTION AT P CLI00210
 NOCONT-THE ERROR FLAG.
 F-NO PROBLEM CLI00220
 T-NO CONTOUR FOUND CLI00230
 ERR-ERROR FLAG RETURNED BY 'NTRSCT'
 F-NO ERROR CLI00240
 T-ITERATION DIVERGED OR MAXIMUM SEARCH AREA EXCEEDED CLI00250

IN ADDITION, IN COMMON ARE...

YMIN-THE LOWER LIMIT ON Y.
 DELTA- THE STEP SIZE, MODIFIED IN THIS SUBROUTINE.
 CON-THE CONTOUR LEVEL.
 RES-THE RESOLUTION LENGTH

OTHER VARIABLES INCLUDE
 GRAD-THE GRADIENT VECTOR.
 P0-THE CURRENT POINT ON THE GRADIENT.
 P1-THE POINT ON THE GRADIENT BEING TESTED
 TO SEE ABOUT CONTOUR EXISTENCE.
 FP0,FP1-THE FUNCTION VALUES OF P0 AND P1.

CALLED SUBROUTINES
 GRAD2-FINDS THE GRADIENT VECTOR OF A FUNCTION AT
 A POINT AND THE SLOPE THERE.
 UNIT-CALCULATES THE NORM AND MAGNITUDE OF A 2 VECTOR.
 VSUM-VECTOR SUM OF THE FORM C=A+SB WHERE S IS SCALAR
 MULTIPLIER OF B.

EXTERNAL FUNCT
 LOGICAL NOCONT
 DIMENSION GRAD(2),P0(2),P1(2)
 COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
 COMMON/LIMIT/YMIN,FMIN
 NOCONT=.FALSE.
 ONEM=-1.0
 IF (NSEARCH.EQ.0)GO TO 7
 DELTA=SIGN(DELTA,FLOAT(NSEARCH))
 FP1=FUNCT(P1(1),P1(2))
 IF(FP1.LT.CON)GO TO 25
 GO TO 22
 3 CONTINUE
 P0(1)=P1(1)
 P0(2)=P1(2)
 FP0=FP1
 C ** FINDING THE UNIT GRADIENT AND THE NEXT POINT ALONG IT.
 4 CALL GRAD2(P0,FUNCT,RES,GRAD,DFDS)

```

      5 CALL VSUM(P0,GRAD,DELTA,P1)           CLI00710
C ** IS THE POINT HEADING BELOW YMIN **
      IF(P1(2).GE.YMIN)GO TO 7               CLI00720
      P1(2)=YMIN                           CLI00730
      CALL VSUM(P1,P0,ONEM,GRAD)            CLI00740
      CALL UNIT(GRAD,GRAD,DELTA)           CLI00750
      IF(ABS(DELTA),LT,RES)GO TO 25        CLI00760
      7 FP1=FUNCT(P1(1),P1(2))             CLI00770
C ** HAS THE CONTOUR BEEN CROSSED **
      8 IF(FP1.GE.CON)GO TO 22              CLI00790
      IF(FP1.GT.FP0)GO TO 3                 CLI00800
      DELTA=DELTA/2.                         CLI00820
      IF(ABS(DELTA),LT,RES)GO TO 25        CLI00830
      GO TO 5                               CLI00840
      25 NOCONT=.TRUE.                     CLI00850
      GO TO 99                             CLI00860
      22 CONTINUE
C BEGIN HORIZONTAL SEARCH
      P0(2)=P1(2)                         CLI00880
      31 P0(1)=P1(1)                         CLI00890
      FP0=FP1                             CLI00900
      40 P1(1)=P0(1)+DELTA                CLI00910
      FP1=FUNCT(P1(1),P1(2))              CLI00920
      IF(ABS(DELTA),LT,RES/2.)GO TO 99    CLI00930
      IF(FP1.GE.CON)GO TO 31              CLI00940
      DELTA=DELTA/2.                       CLI00950
      GO TO 40                            CLI00960
      99 CONTINUE
      RETURN
      END

```

SUBROUTINE CLDIM(CNTRD, HEIGHT, CENWTH, SPCHT, SPCWTH, NCPTS, CPTS5,
1 ERR)

PURPOSE

CLDIM CALCULATES FIVE CONTOUR POINTS AND CLOUD DIMENSIONS AS
SEEN FROM THE SPECIFIED OBSERVER POSITION. CLDIM REQUIRES CLOUD
PARAMETERS FROM THE BUOYANT RISE STAGE OF CLOUD DEVELOPMENT WHICH
ARE SUPPLIED IN COMMON STORAGE /BUOYCL/ AND /PRTINF/ AS WELL AS
VIEWING GEOMETRY WHICH IS SUPPLIED IN COMMON /GEOM/. SPCHT IS
REQUIRED INPUT IN THE ARGUMENTS. ALL OUTPUTS ARE ARGUMENTS.

INPUT

SPCHT THE SPECIFIED HEIGHT AT WHICH THE WIDTH OF THE CLOUD
IS DESIRED. (METERS)

OUTPUT

CNTRD A SINGLY DIMENSIONED ARRAY OF LENGTH 2 WHICH CONTAINS
THE HORIZONTAL AND VERTICAL COORDINATES, RESP., OF THE
CLOUD CENTROID. (METERS) CLD00250
HEIGHT THE HEIGHT OF THE CLOUD IN METERS CLD00270
CENWTH THE WIDTH OF THE CLOUD AT THE CENTROID HEIGHT IN METERS CLD00290
SPCWTH THE WIDTH OF THE CLOUD AT THE SPECIFIED HEIGHT (METERS) CLD00300
NCPTS THE NUMBER OF CONTOUR POINTS (=6) CLD00310
CPTS A DOUBLY DIMENSIONED ARRAY OF SIZE (2,N), N, GE, 5, WHICH
CONTAINS THE HORIZONTAL AND VERTICAL COORDINATES OF
THE FIVE CONTOUR POINTS. (METERS) CLD00320

REQUIRED SUBROUTINES

CLIMB DETERMINES IF THE CONTOUR EXISTS, IF SO FINDS A POINT
ON THE CONTOUR.

CALLED BY DUSTCL

DIMENSION CNTRD(2),CPTS5(2,6),TOP(2) CLD00450
LOGICAL HORIZ,NOCONT,SWITCH,CHANGE,ERR CLD00460
REAL KZ,KX CLD00470
COMMON /BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM CLD00490
COMMON /PRTINF/R0,VGRAY(3),NPRTS CLD00500
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2) CLD00510
COMMON /MODE/ HORIZ CLD00520
COMMON /CLOCK/ T,TWIND CLD00530
COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF CLD00540
COMMON /WNDFRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV CLD00550
COMMON /TRAN/YTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE CLD00560
COMMON /SIG/SIG02,SIGC CLD00570
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU CLD00580
COMMON /DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20), CLD00590
1 QDSC(20,3) CLD00600
COMMON /SPECS/ RES,STEP,TANT,CON CLD00610
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK CLD00620
EXTERNAL FUNCT,GFUN CLD00630
DATA RES,TANT / .4,.1/ CLD00640
HORIZ=.TRUE. CLD00650
ERR=.FALSE. CLD00660
CON=ALOG(VISEXT) CLD00670
CPTS5(2,1)=SPCHT CLD00680
CPTS5(2,6)=SPCHT CLD00690
U=U0*CPTS5(2,1)**UM CLD00700

```

CPTSS5(1,1)=T*U*SINTH          CLD00710
CPTSS5(1,6)=CPTSS5(1,1)        CLD00720
NSERCH=-1                      CLD00730
STEP=20.                        CLD00740
CLD00750
C CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN ONE DIRECTION AT CLD00760
C THE HEIGHT OF THE OBSERVER. CLD00770
CLD00780
C CALL CLIMB(FUNCT,GFUN,CPTSS5,FP1,NSERCH,NOCONT) CLD00790
NSERCH=1                         CLD00800
STEP=20.                         CLD00810
CLD00820
C CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN THE OPPOSITE CLD00830
C DIRECTION AT THE HEIGHT OF THE OBSERVER. CLD00840
CLD00850
CALL CLIMB(FUNCT,GFUN,CPTSS5(1,6),FP1,NSERCH,NOCONT) CLD00860
SPCWTH=CPTSS5(1,6)-CPTSS5(1,1) CLD00870
NCPTS=6                          CLD00880
IF(T.LE.TWIND)GO TO 50           CLD00890
CNTRD(1)=(XTR+YTR*(T-TTR))*SINTH+CENDIF CLD00900
CNTRD(2)=ZTR                     CLD00910
SIGX2=SIG02+2.*KX*(T-TTR)       CLD00920
SIGZ2=SIG02+2.*KZ*(T-TTR)       CLD00930
SIGX=SQRT(SIGX2)                CLD00940
SIGZ=SQRT(SIGZ2)                CLD00950
BOT=1./2.*VISEXT                CLD00960
ARG=BOT*QPUFF(1)/PI/SIGX/SIGZ CLD00970
IF(ARG.LT.1.0)GO TO 998          CLD00980
TOP(1)=CNTRD(1)                 CLD00990
TOP(2)=ZTR+SIGZ*SQRT(2.* ALOG(ARG)) CLD01000
RAD=SIGX*SQRT(2.* ALOG(ARG))   CLD01010
CENWTH=2.*(RAD+PRJARY)          CLD01020
HEIGHT=TOP(2)                   CLD01030
GO TO 100                        CLD01040
50 CNTRD(1)=XCM*SCRN(1)+YCM*SCRN(2)+CENDIF CLD01050
CNTRD(2)=ZCM                     CLD01060
TOP(1)=XTOP*SCRN(1)+YTOP*SCRN(2)+CENDIF CLD01070
TOP(2)=ZCM+RSPH                  CLD01080
IF(TOP(2).GT.ZINV)TOP(2)=ZINV    CLD01090
HEIGHT=TOP(2)                   CLD01100
CENWTH=2.*(RSPH+PRJARY)          CLD01110
100 CPTSS5(1,2)=CNTRD(1)-CENWTH/2. CLD01120
CPTSS5(2,2)=CNTRD(2)             CLD01130
CPTSS5(1,3)=TOP(1)-PRJARY      CLD01140
CPTSS5(2,3)=TOP(2)               CLD01150
CPTSS5(1,4)=TOP(1)+PRJARY      CLD01160
CPTSS5(2,4)=TOP(2)               CLD01170
CPTSS5(1,5)=CNTRD(1)+CENWTH/2. CLD01180
CPTSS5(2,5)=CNTRD(2)             CLD01190
NCPTS=6                          CLD01200
GO TO 999                         CLD01210
998 WRITE(I00UT,1000)              CLD01220
1000 FORMAT(50H *** UPPER PART OF CLOUD HAS DISSIPATED *** ) > CLD01230
999 RETURN                         CLD01240
END                               CLD01250

```

```

C SUBROUTINE COMPC1(NEWATM,NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,
1 PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,
2 NWL,TRNCOR,RECCOR,TIME,DHDT,TRNLOS,NERR) COM00160
C CONTROLING ROUTINE FOR PRECOMPUTING A SINGLE , EXPLOSION PRODUCED COM00170
C DUST CLOUD AND STORING IT ON AN EXTERNAL FILE TO BE USED AT A COM00180
C LATER RUNNING OF THE CODE FOR A RANDOM DISTRIBUTION IN SPACE AND TIME COM00190
C OF CHARGES COM00200
C INPUTS COM00210
C FOR DETAILS SEE DRTRAN COM00220
C OUTPUTS COM00230
C TRNLOS -CALCULATE TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT COM00240
C ****
C ***** DIMENSION ZTEMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3), COM00250
1 RECCOR(3),TRNFRM(2,2),SIDE1(2),SIDE2(2),NCHS(2),ORIG(2),TRN(3), COM00260
2 RECK(3),SRCBAS(2),PAS(6) COM00270
LOGICAL DHDT,NEWATM,NEWSRC,ERR,FLAG COM00280
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2) COM00290
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIR TU,NCLINT,KSTOR,NPLOTU COM00300
COMMON/VL/VLOAD COM00310
COMMON/OPTION/IOPt,IFILE COM00320
COMMON/CLOCK/FTIME,TWIND COM00330
COMMON/BUOYCL/Y(8),SPHNS(3),RTIM COM00340
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3) COM00350
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT, COM00360
+ COOR(2,200),TSTAG(200), COM00370
+ ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25), COM00380
1 RB(3,25),Z2(3,25) COM00390
COMMON/WNDPRM/DXZ0,DYZ0,DZ0,U0,UM,DN,ZINV COM00400
COMMON/CARB/RCARB1,RCARB2 COM00410
COMMON /CONST/PI,PI2,PIRAD,TWOPi,TORRMB,CDEGK COM00420
DATA FLAG/.TRUE./, COM00430
DATA PAS/4HA ,4HB ,4HC ,4HD ,4HE ,4HF / COM00440
IF(IOPt.EQ.4)GO TO 500 COM00450
C PRECOMPUTE INFORMATION AND STORE ON FORTRAN UNIT IFILE COM00460
C THETAX=THWND*PI/180. COM00470
C TWIND=1.ES COM00480
C TTR=1.E5 COM00490
C IF(.NOT.NEWATM)GO TO 10 COM00500
C CALL ATMCAL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR) COM00510
C IF(.NOT.ERR)GO TO 10 COM00520
C NERR=? COM00530
C GO TO 999 COM00540
10 CONTINUE COM00550
C IF(.NOT.NEWSRC)GO TO 20 COM00560
C CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD) COM00570
20 CONTINUE COM00580
C COMPUTE INITIAL LOADING COM00590
C SUM=0.0 COM00600
DO 25 I=1,NDSCS COM00610
SUM=SUM+QDSC(I,1) COM00620
25 CONTINUE COM00630
VLOAD=SPHNS(1)+SUM COM00640
C CALL PRECL TO COMPUTE AND STORE THE QUADRATIC FITS NECESSARY COM00650
FOR THE CONE COM00660
C 1 CALL PRECL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,PHI,DHDT,
CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT) COM00670
1 GO TO 999 COM00680
500 CONTINUE C 100690

```

```

IF(.NOT.FLAG) GO TO 35
READ( IFILE )NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)
READ( IFILE )DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV
READ( IFILE )ICOUNT
DO 30 J=1,ICOUNT
  READ( IFILE ) TIMES(J),(RT(I,J),RBC(I,J),Z2(I,J),XC0(I,J),
  1           XC1(I,J),I=1,3)
30 CONTINUE
WRITE( IOOUT,800 )
800 FORMAT( /,5X,85H ATMOSPHERIC, CHARGE, AND SOIL CHARACTERISTICS USED
1 WHEN THE CLOUD WAS PRECOMPUTED. ) COM00800
C   WRITE OUT ATMOSPHERIC INFORMATION
C
808 FORMAT(1X)
810 FORMAT(30H PASQUILL CATEGORY ,A4) COM00880
  NIO=1
812 FORMAT(8H HT ,F8.2,7H TEMP ,F8.2,7H HT,F8.2,7H WIND ,
1 F8.2) COM00910
  WRITE( IOOUT,814 )THWND
814 FORMAT(22H WIND DIRECTION ,F8.2) COM00940
  WRITE( IOOUT,819 )PHI
  IF(DHDT)WRITE( IOOUT,816 )
816 FORMAT(40H THE INVERSION LAYER HEIGHT IS GROWING > COM00970
  IF(.NOT.DHDT)WRITE( IOOUT,818 )
818 FORMAT(42H THE INVERSION LAYER HEIGHT IS CONSTANT > COM00990
819 FORMAT(/22H LATITUDE ,F8.2) COM01000
C   WRITE SOIL CHARACTERISTICS
C
820 FORMAT(15H SILT CONTENT ,F5.2) COM01040
821 FORMAT( /,15H SOIL-1 ) COM01050
  IF(NSOIL.EQ.2)WRITE( IOOUT,822 )
822 FORMAT( /,15H SOIL-2 ) COM01070
  WRITE( IOOUT,820 )SILT
  WRITE( IOOUT,823 )DSOD
823 FORMAT(21H DEPTH OF SOD ,F5.2) COM01100
C   WRITE EXPLOSIVE CHARGE CHARACTERISTICS
C
824 FORMAT( /,65H SURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC, TIP
1 ON GROUND ) COM01150
  IF(NCHRG.EQ.2)WRITE( IOOUT,825 )
825 FORMAT( /,25H BARE CHARGE ON SURFACE > COM01170
  IF(NCHRG.EQ.3)WRITE( IOOUT,826 )
826 FORMAT( /,45H 30 DEGREE TILTED TIP AT 0.3 METER DEPTH > COM01200
  IF(NCHRG.EQ.4)WRITE( IOOUT,827 )
827 FORMAT( /,45H 30 DEGREE TILTED TIP AT 0.6 METER DEPTH > COM01220
  IF(NCHRG.EQ.5)WRITE( IOOUT,828 )
828 FORMAT( /,40H HORIZONTAL PROJECTILE ON SURFACE > COM01230
  WRITE( IOOUT,829 )CHWT
829 FORMAT(30H WEIGHT OF CHARGE ,F8.2,5H KG. > COM01240
  WRITE( IOOUT,830 )DETDEP
830 FORMAT(30H DETONATION DEPTH ,F8.2) COM01260
  FLAG=.FALSE.
C   COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER
C   DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN
C   THE WIND DIRECTION.
C
35  CONTINUE
  IF (.NOT.NEATM) GO TO 45
  THETAX=THWND*PI/180.
  TRNFRMC(1,1)=COS(THETAX) COM01360
                                         COM01370

```

```

      TRNFRM(2,2)=TRNFRM(1,1)          COM01380
      TRNFRM(1,2)=SIN(thetaX)           COM01390
      TRNFRM(2,1)=-TRNFRM(1,2)         COM01400
C
C MAKE THE ORIGIN OF THE LOCAL COORDINATE SYSTEM THE FIRST      COM01410
C CHARGE LOCATION THAT WAS INPUT BY THE USER                   COM01420
C
      DO 40 I=1,2          SRCBAS(I)=COOR(I,1)          COM01430
      ORIG(I)=SRCBAS(I)          COM01440
  40 CONTINUE          COM01450
C
C CALL SETUP TO COMPUTE THE ARRAY OF DIFFERENCE VECTORS        COM01460
C
      CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)          COM01470
C
C COMPUTE COORDINATES OF THE TRANSMITTER AND RECEIVER IN THE LOCAL      COM01480
C COORDINATE SYSTEM          COM01490
C
      TRN(3)=TRNCOR(3)          COM01500
      REC(3)=RECCOR(3)          COM01510
      DO 60 I=1,2          DO 60 I=1,2          COM01520
      TRN(I)=0.0          REC(I)=0.0          COM01530
      DO 50 J=1,2          DO 50 J=1,2          COM01540
      TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))          COM01550
      REC(I)=REC(I)+RECCOR(I,J)*(RECCOR(J)-ORIG(J))          COM01560
  50 CONTINUE          COM01570
  60 CONTINUE          COM01580
      CONTINUE          COM01590
C
C 45 CALL PRETRN TO COMPUTE THE TRANSMITTANCE ALONG THE SPECIFIED LINE OF      COM01600
C SIGHT          COM01610
C
      CALL PRETRN(TRN,REC,TIME,TRNL0S)          COM01620
  999 RETURN          COM01630
      END          COM01640
                           COM01650
                           COM01660
                           COM01670
                           COM01680
                           COM01690
                           COM01700
                           COM01710
                           COM01720
                           COM01730

```

```

SUBROUTINE CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,
1XNORM,PLEN)                                              CNL00380
CNL00390
CNL00010
CNL00020
CNL00030
CNL00040
CNL00050
CNL00060
CNL00070
CNL00080
CNL00090
CNL00100
CNL00110
CNL00120
CNL00130
CNL00140
CNL00150
CNL00160
CNL00170
CNL00180
CNL00190
CNL00200
CNL00210
CNL00220
CNL00230
CNL00240
CNL00250
CNL00260
CNL00270
CNL00280
CNL00290
CNL00300
CNL00310
CNL00320
CNL00330
CNL00340
CNL00350
CNL00360
CNL00370
CNL00400
CNL00410
CNL00420
CNL00430
CNL00440
CNL00450
CNL00460
CNL00470
CNL00480
CNL00490
CNL00500
CNL00510
CNL00520
CNL00530
CNL00540
CNL00550
CNL00560
CNL00570
CNL00580
CNL00590
CNL00600
CNL00610
CNL00620
CNL00630
CNL00640
CNL00650
CNL00660
CNL00670
CNL00680
CNL00690
CNL00700

C ROUTINE TO FIND THE LENGTH OF A NON-HORIZONTAL LINE THAT INTERSECTS
A CONE

C INPUTS

U - UNIT VECTOR ALONG LINE CONNECTING THE TRANSMITTER AND           CNL00070
RECEIVER.                                                 CNL00080
TR - TRANSMITTER COORDINATES                                     CNL00090
HTTOP- HEIGHT OF THE TOP OF THE CONE SHAPED PORTION OF THE CLOUD   CNL00100
HTBOT- HEIGHT OF THE BOTTOM OF THE CONE SHAPED PORTION OF THE      CNL00110
CLOUD                                                 CNL00120
XCEN - X POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP    CNL00130
YCEN - Y POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP    CNL00140
RTOP - RADIUS OF THE CONE AT THE TOP                            CNL00150
RBOT - RADIUS OF THE CONE AT THE BOTTOM                         CNL00160
XB - X POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD             CNL00170
YB - Y POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD             CNL00180

C OUTPUT

PLEN - LENGTH OF THE INTERSECTION OF CONE AND THE LINE OF SIGHT   CNL00190
FUNTIONS AND SUBROUTINES

NONE

C ****
DIMENSION U(3),TR(3)
IF(U(3).LT.0.0)GO TO 40
C SET UP BOUNDS SO INTERSECTION OF LINE IS SUCH THAT HTBOT < Z < HTTOP
C
PMIN=(HTBOT-TR(3))/U(3)
PMAX=(HTTOP-TR(3))/U(3)
GO TO 50
40 PMIN=(HTTOP-TR(3))/U(3)
PMAX=(HTBOT-TR(3))/U(3)
50 P1=(HTTOP-HTBOT)/U(3)
P0=(HTBOT-TR(3))/U(3)

C SET UP QUADRATIC TO BE SOLVED
C
DX1=U(1)*P1-XCEN+XB
DY1=U(2)*P1-YCEN+YB
DR=RTOP-RBOT
A=DX1**2+DY1**2-DR**2
DX0=TR(1)+U(1)*P0-XB
DY0=TR(2)+U(2)*P0-YB
B=2.*DX1*DX0+DY1*DY0-DR*RBOT
C=(DX0**2+DY0**2-RBOT**2)
RADIC1=(DX1*RBOT-DX0*DR)**2
RADIC2=(DY1*RBOT-DY0*DR)**2
RADIC3=(DX1*DY0-DY1*DX0)**2

C DETERMINE PATH LENGTH IF THE LINE INTERSECTS THE CONE
C
IF(ABS(A).LT.1.E-20)GO TO 60
RADIC=RADIC1+RADIC2-RADIC3

```

```

IF(RADIC.LT.0.0)GO TO 68          CNL00710
ROOT1=-B/A/2.+SQRT(RADIC)/A      CNL00720
ROOT2=-B/A/2.-SQRT(RADIC)/A      CNL00730
C DETERMINE WHICH POINT IS CLOSEST AND WHICH IS FARTHEST FROM THE CNL00740
C TRANSMITTER                           CNL00750
C TEMP1=P1*ROOT1+P0                 CNL00760
C TEMP2=P1*ROOT2+P0                 CNL00770
C IF(A.LT.0.0) GO TO 57             CNL00780
C IF(TEMP1.GT.TEMP2)GO TO 55       CNL00790
C PFAR=TEMP2                         CNL00800
C PNEAR=TEMP1                         CNL00810
C GO TO 67                           CNL00820
55 PFAR=TEMP1                         CNL00830
C PNEAR=TEMP2                         CNL00840
C GO TO 67                           CNL00850
57 IF(U<3).GT.0.0) GO TO 58       CNL00860
C PNEAR=-1.E20                         CNL00870
C PFAR=AMIN1(TEMP1,TEMP2)           CNL00880
C GO TO 67                           CNL00890
58 PNEAR=AMAX1(TEMP1,TEMP2)         CNL00900
C PFAR=1.E20                          CNL00910
C GO TO 67                           CNL00920
C CONSIDER DEGENERATE CASE          CNL00930
C 60 IF(B.LT.1.E-20)GO TO 68        CNL00940
C IF(P1/B.LT.0.0)GO TO 65           CNL00950
C PFAR=P1*(-C/B)+P0                CNL00960
C PNEAR=0.0                           CNL00970
C GO TO 67                           CNL00980
65 PFAR=XNORM                         CNL00990
C PNEAR=P1*(-C/B)+P0                CNL01000
67 PLEN=AMIN1(PFAR,XNORM,PMAX)-AMAX1(PNEAR,0.0,PMIN) CNL01010
C IF(PLEN.LT.0.0)PLEN=0.0           CNL01020
C GO TO 999                           CNL01030
68 PLEN=0.0                           CNL01040
999 RETURN                           CNL01050
END                                CNL01060
CNL01070
CNL01080
CNL01090
CNL01100

```

```

SUBROUTINE CONVRT(T) COV00010
LOGICAL SWITCH,CHANGE COV00020
REAL KZ,KX,N,MDIF COV00030
COMMON/PRTINF/R0,VGRAY(3),NPRTS COV00040
COMMON/BUOYCL/RSPH,DELT,V2SPH,XCMSPH,YCMSPH,ZCMSPH,XTOP,YTOP, COV00050
1SPHNS(3),RISTIM COV00060
COMMON/CLOCK/TIME,TWIND COV00070
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE COV00080
COMMON/SIG/SIG02,SIGC COV00090
COMMON/IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIR TU,NCLIMT,KSTOR,NPLOTUCOV00100
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2) COV00110
COMMON/LOAD/WAKAL,SPHAL COV00120
COMMON/WNDPRM/DXZ0,DYZ0,DZ0,U0,M,NDIF,ZINY COV00130
COMMON/STARS/USTAR,TSTAR,ZSTAR COV00140
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3 COV00150
COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG COV00160
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK COV00170
C ****
C PURPOSE COV00180
C TO CONVERT THE CURRENT BUOYANT CLOUD INTO A THREE DIMENSIONAL COV00190
C GAUSSIAN PUFF TO BE USED BY THE WIND DISPERSION MODEL. COV00200
C INPUT COV00210
C T TIME IN SECONDS AFTER DETONATION COV00220
C CALLING PROGRAM COV00230
C FUNCTIONS AND SUBROUTINES NEEDED COV00240
C WNDCAL COV00250
C AMOUNT COV00260
C AYRG COV00270
C COV00280
C COV00290
C CALLED BY RISE COV00300
C FUNCTIONS AND SUBROUTINES NEEDED COV00310
C WNDCAL COV00320
C AMOUNT COV00330
C AYRG COV00340
C COV00350
C COV00360
C COV00370
C **** CHANGE=.TRUE. COV00380
C TWIND=T COV00390
C VTR=USTAR*WNDCAL(Z0,ZL,ZCMSPH) COV00400
C KZ=DIFFUS(Z0,ZL,ZCMSPH) COV00410
C KX=KZ*DXZ0 COV00420
C TTR=T COV00430
C XTR=XCMSPH COV00440
C ZTR=ZCMSPH COV00450
C DO 10 IPRTS=1,NPRTS COV00460
C QPUFF(IPRTS)=SPHNS(IPRTS) COV00470
10 CONTINUE COV00480
C VOLSPH=(4./3.)*PI*RSPH**3 COV00490
C CALL AMOUNT(VOLSPH,WAKAL,SPHAL) COV00500
C QPUFF(1)=SPHAL COV00510
C 60 SIG02=(2./3.)*RSPH**2 COV00520
C SIG0=SQRT(SIG02) COV00530
C ZX=5.0 COV00540
C 61 CALL AYRG(ZX,T,QTOT,XBAVRG SIG2X,SIG2Y) COV00550
C SIGX=SQRT(SIG2X) COV00560
C SIGY=SQRT(SIG2Y) COV00570
C COMPUTE PARAMETERS NEEDED FOR LOCAL AND COLUMN DENSITY FOR THE WAKE COV00580
C SOLUTION COV00590
C COV00600
C SIGC=(SQRT(SIGX*SIGY)+SQRT(SIG02))/2. COV00610
C XDIF=XCMSPH-XBAVRG COV00620
C YDIF=YCMSPH COV00630
C ZDIF=ZCMSPH-5.0 COV00640
C TDIF=SIGC**2/KX/2. COV00650
C TDX=2.*KX COV00660
C TDZ=2.*KZ COV00670
C QLOC=WAKAL/4./PI/SQRT(2.) COV00680
C COV00690

```

998 QCQL=WAKAL/4./SQR(T(P1))
RETURN
END

COV00700
COV00710
COV00720

```

FUNCTION CSPHER(X,Y,Z,T)                               CSP00230
C COMPUTE EITHER THE COLUMN DENSITY FOR A GIVEN LINE OF SIGHT OR   CSP00010
C COMPUTE THE LOCAL CONCENTRATION AT X,Y,Z FOR THE SPHERE   CSP00020
C INPUT                                                 CSP00030
X,Y,Z      COORDINATES IN METERS. IF LINE INTEGRAL IS DESIRED,   CSP00040
           Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.   CSP00050
T          THE TIME IN SECONDS AFTER DETONATION   CSP00060
C OUTPUT                                                 CSP00070
           RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND   CSP00080
           THE LINE INTEGRAL OF CONCENTRATION (COLUMN DENSITY) IF HORIZ   CSP00090
           IS .TRUE.   CSP00100
C FUNCTIONS AND SUBROUTINES CALLED
C NONE
*****
REAL M,N,KZ,KX                                     CSP00220
LOGICAL HORIZ,SWITCH,CHANGE,TEST                  CSP00240
COMMON /PRTINF/ RD,VGRAV(3),NPRTS               CSP00250
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTP1,SCRN(2)  CSP00260
COMMON /MODE/ HORIZ                            CSP00270
COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV       CSP00280
COMMON /DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),   CSP00290
1 QDSC(20,3)                                         CSP00300
COMMON /MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,   CSP00310
+ COOR(2,200),TSTAG(200),DMMY(401)             CSP00320
COMMON /TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE  CSP00330
COMMON /SIG/SIG02,SIGC                           CSP00340
COMMON /IOUNIT/I0IN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU  CSP00350
COMMON /LOAD/WAKAL,SPHAL                         CSP00360
COMMON /ACL/CWINDS,CWINDC,CWINDW                CSP00370
COMMON /WAKE/XDIF,YDIF,ZDIF,TDX,TDZ,QLOC,QCOL,XBAYRG  CSP00380
COMMON /LOS/TRC(3),RE(3),UC(3)                   CSP00390
COMMON /CHARGE/NCHG                            CSP00400
COMMON /TRANNY/THRESH,TEST,NWL,NSOIL            CSP00410
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK    CSP00420
CSPHER=0.0                                         CSP00430
CWINDSC=0.0                                         CSP00440
CWINDC=0.0                                         CSP00450
IF(NARY.EQ.3)GO TO 999                          CSP00460
IF((T-TTR).LT.1.E-20)GO TO 999                  CSP00470
C COMPUTE CONTRIBUTION FROM BUOYANT CLOUD AFTER SWITCHING TO THE   CSP00480
C WIND MODEL USING A THREE DIMENSIONAL GAUSSIAN PUFF              CSP00490
C
SIGX2=SIG02+2.*KX*(T-TTR)                         CSP00500
SIGZ2=SIG02+2.*KZ*(T-TTR)                         CSP00510
SIGX=SQRT(SIGX2)                                    CSP00520
SIGZ=SQRT(SIGZ2)                                    CSP00530
ARG2=(Z-ZTR)**2/(2.*SIGZ2)                         CSP00540
IF(ABS(ARG2).GT.30.)GO TO 25                      CSP00550
TERM2=EXP(-ARG2)                                    CSP00560
GO TO 26                                           CSP00570
25 TERM2=0.0                                         CSP00580
26 ARG3=(Z+ZTR)**2/(2.*SIGZ2)                      CSP00590
IF(ABS(ARG3).GT.30.)GO TO 27                      CSP00600
TERM3=EXP(-ARG3)                                    CSP00610
GO TO 28                                           CSP00620
27 TERM3=0.0                                         CSP00630
28 IF(HORIZ)GO TO 50                                CSP00640
C COMPUTE CONCENTRATION AT X,Y,Z                  CSP00650
C

```

```

XC=XTR+VTR*(T-TTR)+DIFF(1,NCHG)          CSP00710
YC=DIFF(2,NCHG)                          CSP00720
ARG1=((X-XC)**2+(Y-YC)**2)/2./SIGX2      CSP00730
IF(ABS(ARG1).GT.30.)GO TO 999             CSP00740
TERM=SPHAL/((2.*PI)**(3./2.))/SIGZ/SIGX2   CSP00750
CWINDC=TERM*EXP(-ARG1)*(TERM2+TERM3)       CSP00760
GO TO 100                                CSP00770
C
CC COMPUTE COLUMN DENSITY
50 TERM=SPHAL/2./PI/SIGX/SIGZ            CSP00780
DO 90 I=1,NCHTOT                         CSP00790
XC=XTR+VTR*(T-TTR)+DIFF(1,I)              CSP00800
YC=DIFF(2,I)                            CSP00810
XY=XC*SCRN(1)+YC*SCRN(2)                 CSP00820
ARG1=(X-XY)*2/2./SIGX2                  CSP00830
IF(ABS(ARG1).GT.30.)GO TO 90             CSP00840
CWNDSC=TERM*EXP(-ARG1)*(TERM2+TERM3)     CSP00850
CWINDC=CWINDC+CWNDSC                   CSP00860
CALL TRNCHK(CWINDS,CWINDW,CWINDC)        CSP00870
IF(TEST)GO TO 999                         CSP00880
90 CONTINUE                               CSP00890
100 CSPHER=CWINDC                         CSP00900
999 RETURN                                CSP00910
END                                     CSP00920
CSP00930
CSP00940
CSP00950

```

FUNCTION CWAKE(X,Y,Z,T)

FUNCTION TO COMPUTE THE LOCAL CONCENTRATION OR COLUMN DENSITY AT X,Y,Z AND TIME T AFTER THE BLAST FOR THE WAKE.

INPUT

X,Y,Z COORDINATES IN METERS. IF THE LINE INTEGRAL IS DESIRED THESE ARE NOT USED AND THE LINE IS SPECIFIED BY THE TRANSMITTER AND RECEIVER COORDINATES AND INFORMATION CALCULATED AT THE TIME THE BUOYANT FIREBALL CONVERTED TO THE WIND MODEL.

T THE TIME IN SECONDS AFTER DETONATION

OUTPUT

RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND THE LINE INTEGRAL OF CONCENTRATION (COLUMN DENSITY) IF HORIZ IS .TRUE.

FUNCTIONS AND SUBROUTINES NEEDED

ERF COMPUTE THE ERROR FUNCTION

```
*****  
REAL M,N,KZ,KX  
LOGICAL HORIZ,SWITCH,CHANGE,TEST  
COMMON/PRTINF/ R0,VGRAV(3),NPRTS  
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)  
COMMON /MODE/ HORIZ  
COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV  
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),  
1 QDSC(20,3)  
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,  
+ COOR(2,200),TSTAG(200),DMMY(401)  
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE  
COMMON/SIG/SIG02,SIGC  
COMMON/IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIR TU,NCLIMT,KSTOR,NPLOTU  
COMMON/LOAD/WAKAL,SPHAL  
COMMON/ACL/CWINDS,CWINDC,CWINDW  
COMMON/WAKE/XDIF,YDIF,ZDIF,TDX,TDZ,QLOC,QCOL,XBAYRG  
COMMON/LOS/TR(3),RE(3),U(3)  
COMMON/CHARGE/NCHG  
COMMON/TRAN NY/THRESH,TEST,NWL,NSOIL  
COMMON /CONST/PI,PI2,PIRAD,TWOFI,TURRMB,CDEGK  
IF(NARY.EQ.3)GO TO 999  
IF((T-TTR).LT.1.E-20)GO TO 999
```

COMPUTE CONTRIBUTION FROM THE WAKE AFTER SWITCHING TO THE WIND MODEL FOR A SINGLE CHARGE

```
CWAKE=0.0  
CWINDW=0.0  
CWINDS=0.0  
SIGX2=TDX*(TDIF+(T-TTR))  
SIGZ2=TDZ*(T-TTR)  
IF(HORIZ)GO TO 210  
XB=XBAYRG+DIFF(1,NCHG)  
YB=DIFF(2,NCHG)
```

COMPUTE THE LOCAL CONCENTRATION

```
A=- $(XDIF^{**2}+YDIF^{**2})/2./SIGX2-(ZDIF^{**2}/2./SIGZ2)$   
XX=X-VTR*(T-TTR)  
B0= $(XDIF*(XX-XB)+YDIF*(Y-YB))/SIGX2$   
B1=B0+ $(ZDIF*(Z-5.))/SIGZ2$ 
```

CWA00290
CWA00300
CWA00020
CWA00030
CWA00040
CWA00050
CWA00060
CWA00070
CWA00080
CWA00090
CWA00100
CWA00110
CWA00120
CWA00130
CWA00140
CWA00150
CWA00160
CWA00170
CWA00180
CWA00190
CWA00200
CWA00210
CWA00220
CWA00230
CWA00240
CWA00250
CWA00260
CWA00270
CWA00280
CWA00300
CWA00320
CWA00330
CWA00340
CWA00350
CWA00360
CWA00370
CWA00380
CWA00390
CWA00400
CWA00410
CWA00420
CWA00430
CWA00440
CWA00450
CWA00460
CWA00470
CWA00480
CWA00490
CWA00500
CWA00510
CWA00520
CWA00530
CWA00540
CWA00550
CWA00560
CWA00570
CWA00580
CWA00590
CWA00600
CWA00610
CWA00620
CWA00630
CWA00640
CWA00650
CWA00660
CWA00670
CWA00680
CWA00690
CWA00700

```

B2=B0- $\langle ZDIF*(Z+5, )/SIGZ2\rangle$ 
C0=- $\langle\langle XX-XB \rangle\rangle**2+\langle Y-YB \rangle\rangle**2\rangle/2./SIGX2$ 
C1=C0- $\langle\langle Z-5, )\rangle\rangle**2/2./SIGZ2\rangle$ 
C2=C0- $\langle\langle Z+5, )\rangle\rangle**2/2./SIGZ2\rangle$ 
RTMA=SQRT(-A)
ARG1=C1-B1**2/4./A
ARG2=-B1/2./RTMA
ARG3=RTMA+ARG2
ARG4=C2-B2**2/4./A
ARG5=-B2/2./RTMA
ARG6=RTMA+ARG5
IF<ARG1.LT.-30,>GO TO 221
TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))
GO TO 222
221 TERM1=0.0
222 CONTINUE
IF<ARG4.LT.-30,>GO TO 223
TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))
GO TO 224
223 TERM2=0.0
224 CWAKE=QLOC/SIGX2/SQRT(SIGZ2)/RTMA*(TERM1+TERM2)
GO TO 999
C COMPUTE COLUMN DENSITY
210 DO 245 J=1,NCHTOT
XB=XBAVRG+DIFF(1,J)
YB=DIFF(2,J)
230 A=- $\langle XDIF*U(2)-YDIF*U(1) \rangle\rangle**2/2./SIGX2-ZDIF**2/2./SIGZ2$ 
TRR=TR(1)-VTR*(T-TTR)
B0=(YDIF*U(1)-XDIF*U(2))* $\langle\langle TR(2)-YB \rangle\rangle*U(1)-(TRR-XB)*U(2)\rangle\rangle/SIGX2$ 
B1=B0+ZDIF*(TR(3)-5, )/SIGZ2
B2=B0-ZDIF*(TR(3)+5, )/SIGZ2
C0=- $\langle\langle TR(2)-YB \rangle\rangle*U(1)-(TRR-XB)*U(2) \rangle\rangle**2/2./SIGX2$ 
C1=C0- $\langle\langle TR(3)-5, )\rangle\rangle**2/2./SIGZ2$ 
C2=C0- $\langle\langle TR(3)+5, )\rangle\rangle**2/2./SIGZ2$ 
RTMA=SQRT(-A)
ARG1=C1-B1**2/4./A
ARG2=-B1/2./RTMA
ARG3=RTMA+ARG2
ARG4=C2-B2**2/4./A
ARG5=-B2/2./RTMA
ARG6=RTMA+ARG5
IF<ARG1.GT.30,>ARG1=30.
IF<ARG1.LT.-30,>GO TO 231
TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))
GO TO 232
231 TERM1=0.0
232 CONTINUE
IF<ARG4.LT.-30,>GO TO 233
TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))
GO TO 234
233 TERM2=0.0
234 ARG=SIGX2*S1GZ2
CWNDSW=< QCOL/SQRT(ARG)/RTMA >*(TERM1+TERM2)
240 CONTINUE
CWINDW=CWINDW+CWNDSW
CALL TRNCHK(CWINDS,CWINDW,CWINDC)
IF<TEST>GO TO 999
245 CONTINUE
CWAKE=CWINDW
999 RETURN
END

```

CWA00710
CWA00720
CWA00730
CWA00740
CWA00750
CWA00760
CWA00770
CWA00780
CWA00790
CWA00800
CWA00810
CWA00820
CWA00830
CWA00840
CWA00850
CWA00860
CWA00870
CWA00880
CWA00890
CWA00900
CWA00910
CWA00920
CWA00930
CWA00940
CWA00950
CWA00960
CWA00970
CWA00980
CWA00990
CWA01000
CWA01010
CWA01020
CWA01030
CWA01040
CWA01050
CWA01060
CWA01070
CWA01080
CWA01090
CWA01100
CWA01110
CWA01120
CWA01130
CWA01140
CWA01150
CWA01160
CWA01170
CWA01180
CWA01190
CWA01200
CWA01210
CWA01220
CWA01230
CWA01240
CWA01250
CWA01260
CWA01270
CWA01280
CWA01290
CWA01300
CWA01310
CWA01320
CWA01330

```

FUNCTION CWIND(X,Y,Z,T)                               CWI00270
PURPOSE                                              CWI00010
TO COMPUTE THE CONCENTRATION AT A POINT OR INTEGRATED ALONG   CWI00020
A HORIZONTAL LINE (SKIRT PORTION OF DUST CLOUD)   CWI00030
INPUT                                                 CWI00040
X,Y,Z      COORDINATES IN METERS. IF LINE INTEGRAL IS DESIRED,   CWI00050
Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.   CWI00060
T          THE TIME IN SECONDS AFTER DETONATION   CWI00070
OUTPUT                                              CWI00080
RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND   CWI00090
THE LINE INTEGRAL OF CONCENTRATION IF HORIZ IS .TRUE.   CWI00100
SUBROUTINES CALLED                                 CWI00110
MOMENT      COMPUTES ZERO ORDER MOMENT AND INTERPOLATES FROM   CWI00120
TABLE OF HIGHER ORDER MOMENTS.   CWI00130
CALLED BY FUNCT,TRNCAL                           CWI00140
*****                                         CWI00150
REAL M,N,KZ,KX                                     CWI00160
LOGICAL HORIZ,SWITCH,CHANGE,TEST,SKIP             CWI00170
DIMENSION REF(2),REF0(2)                         CWI00180
COMMON/PRTINF/ R0,VGRAV(3),NPRTS                CWI00190
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTFI,SCRN(2)   CWI00200
COMMON /MODE/HORIZ                                CWI00210
COMMON /WNDFRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV        CWI00220
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),   CWI00230
1 QDSC(20,3)                                      CWI00240
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,   CWI00250
+ COOR(2,200),TSTAG(200),DMMY(401)               CWI00260
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE   CWI00280
COMMON/SIG/SIG02,SIGC                            CWI00290
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCWI00300
COMMON/LOAD/WAKAL,SPHAL                          CWI00310
COMMON/ACL/CWINDC,CWINDW                         CWI00320
COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAYRG   CWI00330
COMMON/LOS/TR(3),RE(3),U(3)                      CWI00340
COMMON/CHARGE/NCHG                             CWI00350
COMMON/TRANNNY/THRESH,TEST,NWL,NSOIL           CWI00360
COMMON/SKIPIT/SKIP                           CWI00370
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK   CWI00380
COMMON /PRTINF/                                 CWI00390
R0          INITIAL RADIUS OF THE CLOUD IN METERS   CWI00400
VGRAV     SINGLY DIMENSIONED ARRAY. VGRAV(I) IS THE OPTICALLY   CWI00410
WEIGHTED AVERAGE SETTLING VELOCITY FOR PARTICLES IN THE   CWI00420
I SIZE RANGE                                         CWI00430
NPRTS      THE NUMBER OF PARTICLE SIZE RANGES   CWI00440
COMMON /DISCS/                                 CWI00450
NDSCS     THE NUMBER OF DISC SOURCES   CWI00460
TDSC      SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF RELEASE   CWI00470
OF THE DISC SOURCES                               CWI00480
XDSC      SINGLY DIMENSIONED ARRAY CONTAINING THE X COORDINATE   CWI00490
OF THE CENTER OF THE DISC SOURCES               CWI00500
ZDSC      SINGLY DIMENSIONED ARRAY CONTAINING THE Z COORDINATE   CWI00510
OF THE CENTER OF THE DISC SOURCES               CWI00520
R2DSC     SINGLY DIMENSIONED ARRAY CONTAINING THE SQUARE OF THE   CWI00530
RADII OF THE DISC SOURCES                         CWI00540
QDSC      DOUBLY DIMENSIONED ARRAY. QDSC(I,J) IS THE OPTICALLY   CWI00550

```

```

C          WEIGHTED MASS OF PARTICLES OF THE J SIZE RANGE IN THE
I DISC.                                              CWI00710
C          SUM THE CONTRIBUTIONS OF THE DISC SOURCES TO THE      CWI00720
OPTICALLY WEIGHTED CONCENTRATION AT (X,Y,Z,T)           CWI00730
C          CWIND=0,                                              CWI00740
C          CWNDS=0.0                                             CWI00750
C          CWINDS=0.0                                             CWI00760
C          CWINDC=0.0                                             CWI00770
C          CWINDW=0.0                                             CWI00780
C          IF(HORIZ)GO TO 110                                 CWI00790
C          COMPUTE CONCENTRATION AT X,Y,Z (FOR SIMULTANEOUS BURST)   CWI00800
C          DO 100 I=1,NDSOS                                  CWI00810
REF0(1)=XDSC(I)                                         CWI00820
REF0(2)=0.0                                               CWI00830
ROH2=R2DSC(I)                                           CWI00840
H=ZDSC(I)                                                 CWI00850
TOF=T-TDSC(I)                                           CWI00860
DO 90 J=1,NPRTS                                         CWI00870
C          DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z       CWI00880
C          CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)   CWI00890
IF(Q.GT.1.E-10)GO TO 50                                CWI00900
CWNDS=0.0                                                 CWI00910
GO TO 100                                                CWI00920
50 CONTINUE                                              CWI00930
RX2=ROH2+2.*SIGW2                                         CWI00940
RY2=ROH2+2.*SIGP2                                         CWI00950
DO 114 NA=1,2                                            CWI00960
REF(NA)=REF0(NA)+DIFF(NA,NCHG)                           CWI00970
C          CONTINUE                                           CWI00980
ARG=- $(X-REF(1)-XBAR)^2/RX2$                             CWI00990
IF(ABS(ARG).GT.30.)GO TO 100                            CWI01000
CWNDS=(Q/RTPI/SQRT(RX2))*EXP(ARG)                      CWI01010
ARG=- $(Y-REF(2))^2/RY2$                                 CWI01020
IF(ABS(ARG).GT.30.)GO TO 100                            CWI01030
CY=EXP(ARG)                                              CWI01040
CWNDS=CDSCK(I,J)*CWNDS*CY                             CWI01050
CWINDS=CWINDS+CWNDS                                     CWI01060
C          CONTINUE                                           CWI01070
100 CONTINUE                                              CWI01080
CWIND=CWINDS                                           CWI01090
GO TO 999                                                CWI01100
110 DO 220 ICHG=1,NTOT                                    CWI01110
IF(T.LT.TSTAG(ICHG))GO TO 220                         CWI01120
DO 211 I=1,NDSOS                                       CWI01130
TOF=T-TDSC(I)-TSTAG(ICHG)                            CWI01140
REF0(1)=XDSC(I)                                         CWI01150
REF0(2)=0.0                                              CWI01160
ROH2=R2DSC(I)                                           CWI01170
H=ZDSC(I)                                                 CWI01180
IF(HORIZ) REF0(1)=REF0(1)*SINTH                      CWI01190
DO 210 J=1,NPRTS                                       CWI01200
CWNDS=0.0                                                 CWI01210
C          DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z       CWI01220
C          CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)   CWI01230
IF(Q.GT.1.E-10) GO TO 113                            CWI01240
C          IF Q IS TOO SMALL, ITS CONTRIBUTION IS IGNORED    CWI01250
CWNDS=0.0                                                 CWI01260
GO TO 210                                              CWI01270
113 CONTINUE                                              CWI01280
CWI01290
CWI01300
CWI01310
CWI01320
CWI01330
CWI01340
CWI01350
CWI01360
CWI01370
CWI01380
CWI01390
CWI01400

```

```

DO 200 MA=1,NCHTOT
INDEX=MA
IF(NARY.EQ.3)INDEX=ICHG
RX2=R0H2+2.*SIGN2
RY2=R0H2+2.*SIGP2
C COMPUTE CONCENTRATION ALONG LINE OF SIGHT SPECIFIED BY X,Z
120 CONTINUE
REF(1)=REF0(1)+PRSEP(INDX)
REFF2=RX2*SINTH2+RY2*COSTH2
ARG=-((X-REF(1))-XBAR*SINTH)**2/REFF2
IF(ABS(ARG).GT.30.)GO TO 150
CWNDSC=EXP(ARG)/SQRT(REFF2)/RTPI
CWNDSC=CWNDSC*Q*QDSC(I,J)
150 CONTINUE
CWINDS=CWINDS+CWNDSC
IF(SKIP)GO TO 190
CALL TRNCHK(CWINDS,CWINDW,CWINDC)
IF(TEST)GO TO 999
190 CONTINUE
200 CONTINUE
210 CONTINUE
211 CONTINUE
220 CONTINUE
CWIND=CWINDS
999 RETURN
END

```

CWI01410
CWI01420
CWI01430
CWI01440
CWI01450
CWI01460
CWI01470
CWI01480
CWI01490
CWI01500
CWI01510
CWI01520
CWI01530
CWI01540
CWI01550
CWI01560
CWI01570
CWI01580
CWI01590
CWI01600
CWI01610
CWI01620
CWI01630
CWI01640
CWI01650
CWI01660
CWI01670
CWI01680

```

SUBROUTINE DIFEQ(N,T,Y,YP)
REAL KM,KZ,KX
LOGICAL SWITCH,CHANGE
DIMENSION Y(N),YP(N)
COMMON/PRTINF/R0CL,VGRAV(3),NPRTS
COMMON/WNDFRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINY
COMMON /ARRAY/OVRMAP,AREA,PERIM,PRJARY,CENDIF
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+           DMM(600),DMMY(401)
COMMON /BURST/ ACCEL,TBURST
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/EKWIND/ALP,C,PXF,UHAT,VHAT
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NCLIMT,KSTOR,NPLOTUD
DATA ALPHAK/.25/
*****
* PURPOSE
* DIFEQ CONTAINS THE PARTIAL DIFFERENTIAL EQUATIONS FOR THE
* RISE OF A BUOYANT CLOUD WHICH ARE USED BY SUBROUTINE RKM.
*
* INPUT
* N      THE NUMBER OF DEPENDENT VARIABLES
* T      THE INDEPENDENT VARIABLE. I.E. TIME
* Y(1)   RADIUS OF CLOUD
* Y(2)   CLOUD TEMPERATURE MINUS SURROUNDING TEMPERATURE
* Y(3)   VERTICAL VELOCITY OF CLOUD
* Y(4)   X-COORDINATE OF CENTER OF MASS FOR THE CLOUD
* Y(5)   Y-COORDINATE OF CENTER OF MASS FOR THE CLOUD
* Y(6)   THE HEIGHT OF THE CLOUD C.O.M.
* Y(7)   X-COORDINATE OF TOP OF CLOUD
* Y(8)   Y-COORDINATE OF TOP OF CLOUD
*
* OUTPUT
* YP    AN ARRAY CONTAINING COMPUTED DERIVATIVES OF THE DEPENDENT
*       VARIABLES WITH RESPECT TO THE INDEPENDENT VARIABLE.
*
* REQUIRED FUNCTIONS
* TEMP   CALCULATES AMBIENT ATMOSPHERIC TEMPERATURE AND THE
*         TEMPERATURE GRADIENT AT CLOUD HEIGHT.
* WIN    CALCULATES THE WIND SPEED IN THE X AND Y DIRECTION AT
*         CLOUD HEIGHT.
* DIFFUS COMPUTES THE DIFFUSIVITY AT A SPECIFIED HEIGHT.
*
* CALLED BY RKM
* *****
* IF(T.LT.TBURST)GO TO 200
* IF(Y(6).GT.ZSTAR)GO TO 5
* CALL TEMP(Y(6),TA,DTADZ)
* GO TO 6
5  TA=TC1+TC2*Y(6)+TC3*Y(6)**2
* DTADZ=TC2+2.*TC3*Y(6)
6  CALL WIN(Y(6),XWCM,YWCM)
* TOP=Y(6)+Y(1)
* CALL WINK(TOP,XWTOP,YWTOP)

```

```

TDIF=TA-T0          DFQ00710
TA                  THE AMBIENT ATMOSPHERIC TEMPERATURE AT CLOUD HEIGHT DFQ00720
DTADZ               THE TEMPERATURE GRADIENT AT CLOUD HEIGHT DFQ00730
XWCM                THE WIND SPEED IN THE X DIRECTION AT CLOUD C.O.M. DFQ00740
YXCM                THE WIND SPEED IN THE Y DIRECTION AT CLOUD C.O.M. DFQ00750
XWTOP               THE WIND SPEED IN THE X DIRECTION AT THE TOP OF THE CLOUD DFQ00760
YWTOP               THE WIND SPEED IN THE Y DIRECTION AT THE TOP OF THE CLOUD DFQ00770
TR                  THE RATIO OF CLOUD TEMPERATURE TO AMBIENT TEMPERATURE DFQ00780
DFQ00790
TR=Y(2)/TA          DFQ00800
DFQ00810
CALCULATE ARVOL, THE SURFACE AREA TO VOLUME RATIO DFQ00820
ARVOL=3./Y(1)        DFQ00830
DFQ00840
DFQ00850
DFQ00860
DFQ00870
DEFINITION OF DIFFERENTIAL EQUATIONS DFQ00880
DFQ00890
YP(1)=ALPHAK*ABS(Y(3)) DFQ00900
Z21=Y(6)             DFQ00910
KM=DIFFUS(Z0,ZL,Z21) DFQ00920
GROWTH=KM/Y(1)       DFQ00930
IF(YP(1).LT.GROWTH)YP(1)=GROWTH DFQ00940
YP(2)=-1.+TR)*ARVOL*Y(2)+YP(1)-Y(3)*(DTADZ) DFQ00950
YP(3)=9.-8.*TR-1.4*ARVOL*Y(3)*YP(1) DFQ00960
IF(Y(1)+Y(6).GT.ZINV)YP(3)=0.0 DFQ00970
YP(4)=XWCM           DFQ00980
YP(5)=YXCM           DFQ00990
YP(6)=Y(3)            DFQ01000
IF(Y(1)+Y(6).GT.ZINV)YP(6)=0.0 DFQ01010
YP(7)=XWTOP          DFQ01020
YP(8)=YWTOP          DFQ01030
GO TO 999             DFQ01040
200 CONTINUE          DFQ01050
DO 210 I=1,N          DFQ01060
YP(I)=0.              DFQ01070
210 CONTINUE          DFQ01080
YP(3)=ACCEL          DFQ01090
YP(6)=Y(3)            DFQ01100
999 RETURN            DFQ01110
END

```

```

FUNCTION DIFFUS(Z0,ZL,Z)
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/COEF/AW,CW,BW,AT,CT,BT,DT
DIF00010
DIF00020
DIF00030
DIF00040
DIF00050
DIF00060
DIF00070
DIF00080
DIF00090
DIF00100
DIF00110
DIF00120
DIF00130
DIF00140
DIF00150
DIF00160
DIF00170
DIF00180
DIF00190
DIF00200
DIF00210
DIF00220
DIF00230
DIF00240
DIF00250
DIF00260
DIF00270
DIF00280
DIF00290
DIF00300
DIF00310
DIF00320
DIF00330
DIF00340
DIF00350
DIF00360
DIF00370
DIF00380
DIF00390
DIF00400
DIF00410
DIF00420
DIF00430
DIF00440
DIF00450
DIF00460
DIF00470
DIF00480
DIF00490
DIF00500
DIF00510
DIF00520
DIF00530
DIF00540

***** PURPOSE *****

TO CALCULATE THE DIFFUSIVITY AT A GIVEN HEIGHT

INPUTS
Z0  FRICTION HEIGHT IN METERS.
ZL  MONIN OBUKHOV LENGTH IN METERS.
Z  HEIGHT AT WHICH DIFFUSIVITY IS DESIRED.

CALLED BY ATMAL, RISE AND DIFEQ

SUBROUTINES AND FUNCTIONS NEEDED
NONE

***** ZZ=Z
IF(Z.GT.ZSTAR)Z=ZSTAR

NEUTRAL CASE
IF(ABS(ZL).LT.1.E3)GO TO 100
DIFFUS=.4*USTAR*Z
GO TO 999
100 IF(ZL.GT.0.0)GO TO 200

UNSTABLE CASE
S=Z/ZL
IF(S.LT.-2.)GO TO 110
DIFFUS=.4*ABS(ZL*USTAR*S*(1.-16.*S)**(1./4.))
GO TO 999
110 DIFFUS=.4*ABS(ZL*(3./AW)*(-1.*S)**(4./3.))*USTAR
GO TO 999

STABLE CASE
200 S=Z/ZL
S0=Z0/ZL
IF(S.GT.1.5)GO TO 210
DIFFUS=.4*ZL*USTAR*ABS(1./(1./(S0+S)+7.))
GO TO 999
210 DIFFUS=.4*ZL*USTAR*ABS(1./BW)
Z=ZZ
999 RETURN
END

```

C
CC

```
FUNCTION DOTPRD(A,B)
DIMENSION A(2),B(2)
COTPRD IS THE SCALER PRODUCT OF A AND B
DOTPRD=A(1)*B(1)+A(2)*B(2)
RETURN
END
```

```
DOT00010
DOT00020
DOT00030
DOT00040
DOT00050
DOT00060
DOT00070
DOT00080
```

```

FUNCTION DTERPI(NDIM,XI,XVAL,VAL,VMIN,WORK)
*****  

C PURPOSE  

PERFORMS AN N-DIMENSIONAL LINEAR INTERPOLATION  

C INPUT  

NDIM - THE NUMBER OF DIMENSIONS. (- DONT RECALCULATE WEIGHTS)  

XI - THE POINT IN THE HYPERSPACE AT WHICH THE INTERPOLATED  

     VALUE IS DESIRED. XI MUST BE A VECTOR OF ATLEAST NDIM  

     IN LENGTH.  

XVAL - THE COORDINATE VALUES AT THE CORNERS OF THE HYPERCUBE.  

     THE VECTOR MUST BE SET UP LIKE A TWO-DIMENSIONAL ARRAY  

     (2 X NDIM), WHERE THE FIRST SUBSCRIPT REFERS TO THE  

     HYPERCUBE COORDINATES IN THE SECOND SUBSCRIPTS  

VAL - THE FUNCTIONAL VALUES AT THE CORNERS OF THE HYPERCUBE  

     SURROUNDING XI. THIS VECTOR MUST BE FILLED EQUIVALENT  

     TO AN NDIM ARRAY WITH EACH DIMENSION AS 2. THE SIZE  

     OF VAL SHOULD BE ATLEAST 2**NDIM.  

VMIN - A MINIMUM VALUE OF VAL FOR WHICH THE INTERPOLATION  

     WILL USE A CORNER VALUE.  

WORK - A WORK VECTOR OF ATLEAST NDIM*2. USE TO STORE COOR-  

     DINATE WEIGHTS.  

C OUTPUT  

     RETURNS INTERPOLATED VALUE OF VAL AT XI  

     CALLED BY MOMENT  

*****  

DIMENSION XI(4),XVAL(8),VAL(16),WORK(8)  

C SET UP THE COORDINATE WEIGHTS  

NDI=IABS(NDIM)  

IF(NDIM .LT. 0) GO TO 1  

DO 100 I=1,NDI  

I2=I*2  

I1=I2-1  

WORK(I2)=<(XI(I)-XVAL(I1))/(XVAL(I2)-XVAL(I1))>  

WORK(I1)=1. - WORK(I2)  

CONTINUE  

100 C INTERPOLATE - USE BINARY COUNTER FOR COORDINATE LOCATION  

1      DTERPI=0.  

SUM=0.  

ND=2**NDI  

DO 201 I=1,ND  

IF(VAL(I) .LT. VMIN) GO TO 201  

L=I-1  

WEIGHT=1.  

DO 200 J=1,NDI  

N=MOD(L,2) + J*2 - 1  

WEIGHT=WEIGHT*WORK(N)  

L=L/2  

CONTINUE  

200 SUM=SUM + WEIGHT  

DTERPI=DTERPI + WEIGHT*VAL(I)  

CONTINUE  

201 IF(SUM .EQ. 0,) GO TO 202  

DTERPI=DTERPI/SUM  

RETURN

```

C
202 STOP
END

DTI00710
DTI00720
DTI00730

(

SUBROUTINE DTERPS(II,X,VAL,NZ)
DIMENSION X(81,4,3),VAL(16),II(4)

DTS00010
DTS00020
DTS00030
DTS00040
DTS00050
DTS00060
DTS00070
DTS00080
DTS00090
DTS00100
DTS00110
DTS00120
DTS00130
DTS00140
DTS00150
DTS00160
DTS00170
DTS00180
DTS00190
DTS00200
DTS00210
DTS00220
DTS00230
DTS00240
DTS00250
DTS00260
DTS00270
DTS00280
DTS00290
DTS00300
DTS00310
DTS00320
DTS00330
DTS00340
DTS00350
DTS00360
DTS00370
DTS00380
DTS00390
DTS00400
DTS00410
DTS00420
DTS00430
DTS00440
DTS00450
DTS00460
DTS00470
DTS00480

PURPOSE

TO SET UP A ONE DIMENSIONAL ARRAY OF THE VALUES CORRESPONDING
TO THE CORNERS OF THE CUBE WITHIN A TABULATED ARRAY WITH
LOWEST CORNER INDICES GIVEN

INPUT

II SINGLY DIMENSIONED ARRAY CONTAINING THE INDICES OF THE
 LOWEST CORNER OF THE CUBE
X A TRIPLY DIMENSIONED ARRAY CONTAINING THE TABULATED
 VALUES TO BE SET UP. THE FIRST INDEX IS THE COLLAPSED
 INDEX FOR THE FIRST TWO INDICES OF A FOUR-DIMENSIONAL
 ARRAY
NZ THE RANGE OF THE FIRST INDEX OF THE FOUR-DIMENSIONAL
 ARRAY

OUTPUT

VAL SINGLY DIMENSIONED ARRAY CONTAINING THE VALUES OF X
 FOR THE 16 CORNER POINTS OF THE CUBE

CALLED BY MOMENT

M=0
DO 104 L=1,2
LX=L + II(4) - 1
DO 103 K=1,2
KX=K + II(3) - 1
DO 102 JI=1,2
JIX=(JI + II(2) - 2)*NZ
DO 101 IJ=1,2
IJX=JIX + IJ + II(1) - 1
M=M+1
VAL(M)=X(IJX,KX,LX)
CONTINUE
102 CONTINUE
103 CONTINUE
104 CONTINUE
RETURN
END

```

SUBROUTINE DUSTCL(NEWATH, NATHOS, ZTMP, TMPMES, ZWND, WNDMES, PHI,
1 THWND, NEWSRC, CHWT, NCHRG, DETDEP, NSOIL, DSOD, DUS00010
2 LOSTRN, TRNCOR, RECCOR, EDGE, OBSCOR, SPCHT, NEWTIM, DUS00020
3 TIME, TRHLOS, CNTRD, HEIGHT, CENWTH, SPCWTH, NCPTS, CPTS, NERR, DUS00030
4 NCHS, SRCBAS, SIDE1, SIDE2, DHDT) DUS00040
LOGICAL NEWATM, NEWSRC, LOSTRN, EDGE, NEWTIM, HORIZ, ERR DUS00050
LOGICAL SWITCH, CHANGE, DHDT DUS00060
DIMENSION ZTMP(2), TMPMES(2), ZWND(2), WNDMES(2), TRNCOR(3) DUS00070
1 , RECCOR(3), CPTS(2,6), ORIG(2), TRNFRM(2,2), TRN(3), REC(3) DUS00080
CNTRD(2), OBSCOR(2), DIR(2) DUS00090
3 . SRCBAS(2), SIDE1(2), SIDE2(2), TEMP(2), NCHS(2) DUS00100
REAL KZ, KX DUS00110
COMMON /GEOM/COSTH2, SINTH, SINTH2, VISEXT, RTPI, SCRN(2) DUS00120
COMMON /MODE/ HORIZ DUS00130
COMMON /WNDPRM/ DX20, DYX0, DZ0, U0, UM, DN, ZINV DUS00140
COMMON /CLOCK/ FTIME, TWIND DUS00150
COMMON/MOS/DIFF(2,200), NCHTOT, PRSEP(200), NTOT, NARY, ITOT, DUS00160
+ COOR(2,200), TSTAG(200), DMMY(401) DUS00170
COMMON /ARRAY/OVRLAP, AREA, PERIM, PRJARY, CENDIF DUS00180
COMMON /IOUNIT/IQIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUDUS00190
COMMON/RCARB/RCARB1, RCARB2 DUS00200
COMMON/ACL/CWINDS, CWINDC, CWINDW DUS00210
COMMON/TRAN/VTR, KZ, KX, TTR, XTR, ZTR, QPUFF(3), SWITCH, CHANGE DUS00220
COMMON /CONST/PI, PI2, PIRAD, TWOP, TORRMB, CDEGK DUS00230
DATA ONEM/-1. / DUS00240
DUS00250
*****
```

PURPOSE

DUSTCL CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES
THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE,
EXPLOSIVE CHARACTERISTICS, AND WAVELENGTH.

SEE COMMENTS IN DRTRAN FOR DETAILS.

SUBROUTINES CALLED

ATMCAL	ACCEPTS METEOROLOGICAL DATA AS ARGUMENTS AND COMPUTES NECESSARY PARAMETERS IN COMMON /WNDPRM/, /EKTEMP/ /STARS/	DUS00390
SOURCE	ACCEPTS SOIL, CHARGE, AND WAVELENGTH SPECIFICATIONS AS INPUT AND COMPUTES NECESSARY PARAMETERS AND INITIAL VALUES IN COMMON /PRTINF/, /BUOYCL/ AND /CARB/	DUS00400
SETUP	ACCEPTS THE USER DEFINED COORDINATES OF THE CHARGES AND CONVERTS THEM TO THE INTERNAL (LOCAL) COORDINATE SYSTEM. ALSO COMPUTES THE AREA AND PERIMETER OF THE BOUNDING PARALLELOGRAM AND OVERLAP DISTANCE OF THE CHARGES AND RETURNS THEM IN COMMON /ARRAY/ AND /SEPRTH/.	DUS00410
RISE	GIVEN CLOUD DIMENSIONS DURING BUOYANT RISE DEVELOPMENT OF CLOUD, RISE CALCULATES THE DIMENSIONS AT A LATER TIME	DUS00420
CLDIM	DETERMINES THE OUTPUT VARIABLES DESCRIBING THE CLOUD DIMENSIONS.	DUS00430
TRNCAL	CONTROLLING ROUTINE FO THE CALCULATION OF TRANSMITTANCES.	DUS00440

```

*****  

IF(LOSTRN.OR.EDGE)GO TO 101 DUS00450
NERR=4 DUS00460
GO TO 999 DUS00470
101 IF(.NOT.NEWATM) GO TO 200 DUS00480
THETAX=THWND*PIRAD DUS00490
CALL ATMCAL(NATHOS, ZTMP, TMPMES, ZWND, WNDMES, PHI, THETAX, DHDT, ERR) DUS00500
99999 IF(.NOT.ERR)GO TO 155 DUS00510
```

```

NERR=7          DUS00710
GO TO 999      DUS00720
155 CONTINUE    DUS00730
               DUS00740
               DUS00750
               DUS00760
               DUS00770
               DUS00780
               DUS00790
               DUS00800
               DUS00810
               DUS00820
               DUS00830
               DUS00840
               DUS00850
               DUS00860
               DUS00870
               DUS00880
               DUS00890
               DUS00900
               DUS00910
               DUS00920
               DUS00930
               DUS00940
               DUS00950
               DUS00960
               DUS00970
               DUS00980
               DUS00990
               DUS01000
               DUS01010
               DUS01020
               DUS01030
               DUS01040
               DUS01050
               DUS01060
               DUS01070
               DUS01080
               DUS01090
               DUS01100
               DUS01110
               DUS01120
               DUS01130
               DUS01140
               DUS01150
               DUS01160
               DUS01170
               DUS01180
               DUS01190
               DUS01200
               DUS01210
               DUS01220
               DUS01230
               DUS01240
               DUS01250
               DUS01260
               DUS01270
               DUS01280
               DUS01290
               DUS01300
               DUS01310
               DUS01320
               DUS01330
               DUS01340
               DUS01350
               DUS01360
               DUS01370
               DUS01380
               DUS01390
               DUS01400

COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT USER
DEFINED COORDINATES INTO LOCAL COORDINATES WITH X AXIS IN
THE WIND DIRECTION.                                         DUS00710
DUS00720
DUS00730
DUS00740
DUS00750
DUS00760
DUS00770
DUS00780
DUS00790
DUS00800
DUS00810
DUS00820
DUS00830
DUS00840
DUS00850
DUS00860
DUS00870
DUS00880
DUS00890
DUS00900
DUS00910
DUS00920
DUS00930
DUS00940
DUS00950
DUS00960
DUS00970
DUS00980
DUS00990
DUS01000
DUS01010
DUS01020
DUS01030
DUS01040
DUS01050
DUS01060
DUS01070
DUS01080
DUS01090
DUS01100
DUS01110
DUS01120
DUS01130
DUS01140
DUS01150
DUS01160
DUS01170
DUS01180
DUS01190
DUS01200
DUS01210
DUS01220
DUS01230
DUS01240
DUS01250
DUS01260
DUS01270
DUS01280
DUS01290
DUS01300
DUS01310
DUS01320
DUS01330
DUS01340
DUS01350
DUS01360
DUS01370
DUS01380
DUS01390
DUS01400

200 CONTINUE
IF( .NOT. NEWSRC) GO TO 300
TWIND=1.E5
TTR=1.E5
TPRES=0.
DEL=.001
DO 250 I=1,2
IF(NARY.GT.1)SRCBAS(I)=COOR(I,1)
ORIG(I)=SRCBAS(I)
250 CONTINUE
CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)
CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)
300 CONTINUE
IF( .NOT. LOSTRN) GO TO 400
CONVERT TRNCOR AND RECCOR TO LOCAL COORDINATES WITH ORIGIN AT
SRCBAS AND X AXIS IN WIND DIRECTION.                                         DUS00710
DUS00720
DUS00730
DUS00740
DUS00750
DUS00760
DUS00770
DUS00780
DUS00790
DUS00800
DUS00810
DUS00820
DUS00830
DUS00840
DUS00850
DUS00860
DUS00870
DUS00880
DUS00890
DUS00900
DUS00910
DUS00920
DUS00930
DUS00940
DUS00950
DUS00960
DUS00970
DUS00980
DUS00990
DUS01000
DUS01010
DUS01020
DUS01030
DUS01040
DUS01050
DUS01060
DUS01070
DUS01080
DUS01090
DUS01100
DUS01110
DUS01120
DUS01130
DUS01140
DUS01150
DUS01160
DUS01170
DUS01180
DUS01190
DUS01200
DUS01210
DUS01220
DUS01230
DUS01240
DUS01250
DUS01260
DUS01270
DUS01280
DUS01290
DUS01300
DUS01310
DUS01320
DUS01330
DUS01340
DUS01350
DUS01360
DUS01370
DUS01380
DUS01390
DUS01400

TRN(3)=TRNCOR(3)
REC(3)=RECCOR(3)
DO 320 I=1,2
TRN(I)=0.
REC(I)=0.
DO 310 J=1,2
TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))
REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))
310 CONTINUE
320 CONTINUE
400 CONTINUE
IF( .NOT. EDGE) GO TO 500
COMPUTE A UNIT VECTOR IN THE DIRECTION OF THE OBSERVERS LINE
OF SIGHT AND A UNIT VECTOR PERPENDICULAR TO THE LINE OF SIGHT
CALL VSUM(ORIG,OBSCOR,ONEM,DIR)
CALL UNIT(DIR,DIR,RANGE)
COSTH=0.
SINTH=0.
DO 410 J=1,2
COSTH=COSTH+TRNFRM(1,J)*DIR(J)
SINTH=SINTH+TRNFRM(2,J)*DIR(J)
410 CONTINUE
SINTH2=SINTH*SINTH
COSTH2=COSTH**2
SCRN(1)=SINTH
SCRN(2)=-COSTH
COMPUTE THE PROJECTION OF EACH DIFFERENCE VECTOR DIFF ONTO THE
VECTOR PERPENDICULAR TO THE LINE OF SIGHT,(DIFF(1,J),DIFF(2,J))
IS THE VECTOR FROM THE REFERENCE CHARGE TO THE JTH CHARGE
LOCATION IN THE INTERNAL COORDINATE SYSTEM.                                         DUS00710
DUS00720
DUS00730
DUS00740
DUS00750
DUS00760
DUS00770
DUS00780
DUS00790
DUS00800
DUS00810
DUS00820
DUS00830
DUS00840
DUS00850
DUS00860
DUS00870
DUS00880
DUS00890
DUS00900
DUS00910
DUS00920
DUS00930
DUS00940
DUS00950
DUS00960
DUS00970
DUS00980
DUS00990
DUS01000
DUS01010
DUS01020
DUS01030
DUS01040
DUS01050
DUS01060
DUS01070
DUS01080
DUS01090
DUS01100
DUS01110
DUS01120
DUS01130
DUS01140
DUS01150
DUS01160
DUS01170
DUS01180
DUS01190
DUS01200
DUS01210
DUS01220
DUS01230
DUS01240
DUS01250
DUS01260
DUS01270
DUS01280
DUS01290
DUS01300
DUS01310
DUS01320
DUS01330
DUS01340
DUS01350
DUS01360
DUS01370
DUS01380
DUS01390
DUS01400

PARY1=0.0
PARY2=0.0
DO 420 J=1,NCHTOT
DO 415 I=1,2
TEMP(I)=DIFF(I,J)

```

```

415 CONTINUE          DUS01410
  PRSEP(J)=DOTPRD(TEMP,SCRN)
  X=PRSEP(J)           DUS01420
  IF(X.LT.0.0)GO TO 416 DUS01430
  IF(X.GT.PARY1)PARY1=X DUS01440
  GO TO 420           DUS01450
416 IF(X.LT.PARY2)PARY2=X DUS01460
420 CONTINUE          DUS01470
  PRJARY=(PARY1-PARY2)/2 DUS01480
  CENDIF=(PARY1+PARY2)/2 DUS01490
500 CONTINUE          DUS01500
  IF(NARY.EQ.3)GO TO 600 DUS01510
  IF(NEWTIME) CALL RISE(TPRES,TIME,DEL) DUS01520
600 IF(.NOT.EDGE) GO TO 650 DUS01530
  FTIME=TIME           DUS01540
  CALL CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS,ERR) DUS01550
  IF(.NOT.ERR)GO TO 650 DUS01560
  NERR=6               DUS01570
  GO TO 999             DUS01580
650 CONTINUE          DUS01590
  IF(.NOT.LOSTRN)GO TO 999 DUS01600
C   CALL TRNCAL(TRN,REC,TIME,TRNLOS) DUS01610
999 RETURN            DUS01620
  END                 DUS01630
                           DUS01640
                           DUS01650

```

```

FUNCTION ERF(X)
CALCULATES THE ERROR FUNCTION
      INPUT
        X    VALUE AT WHICH ERROR FUNCTION IS DESIRED
      FUNCTIONS AND SUBROUTINES NEEDED
        NONE
*****
DIMENSION P(3,5),Q(3,5)
DATA RPI/.5641896/
DATA P/2.138533E+01,7.373888E+00,-4.257996E-02,
+     1.722276E+00,6.865018E+00,-1.960690E-01,
+     3.166529E-01,3.031799E+00,-5.168823E-02,
+     0.,5.631696E-01,0.,
+     0.,4.318779E-05,0./
DATA Q/1.895226E+01,7.373961E+00,1.509421E-01,
+     7.843746E+00,1.518491E+01,9.214524E-01,
+     1.000000E+00,1.279553E+01,1.000000E+00,
+     0.,5.354217E+00,0.,
+     0.,1.000000E+00,0./
AX=ABS(X)
ERFC=0.0
IF(AX.GT.11.0) GO TO 300
X2=AX*AX
I=2
IF(AX.LT.0.5) I=1
IF(AX.GT.4.) I=3
IF(I-2) 10,20,30
10 N=3
Z=X2
GO TO 40
20 N=5
Z=AX
GO TO 40
30 N=3
Z=1./X2
40 SP=P(I,N)
SQ=Q(I,N)
N1=N-1
DO 50 K=1,N1
J=N-K
SP=SP*Z+P(I,J)
50 SQ=SQ*Z+Q(I,J)
IF(I-2) 60,70,80
60 ERFC=1.0-X*SP/SQ
ERF=1.-ERFC
RETURN
70 ERFC=EXP(-X2)*SP/SQ
GO TO 300
80 ERFC=EXP(-X2)/AX*(RPI+SP/(SQ*X2))
300 IF(X.LT.0.0) ERFC=2.0-ERFC
ERF=1.-ERFC
RETURN
END

```

ERF00120
 ERF00010
 ERF00020
 ERF00030
 ERF00040
 ERF00050
 ERF00060
 ERF00070
 ERF00080
 ERF00090
 ERF00100
 ERF00110
 ERF00130
 ERF00140
 ERF00150
 ERF00160
 ERF00170
 ERF00180
 ERF00190
 ERF00200
 ERF00210
 ERF00220
 ERF00230
 ERF00240
 ERF00250
 ERF00260
 ERF00270
 ERF00280
 ERF00290
 ERF00300
 ERF00310
 ERF00320
 ERF00330
 ERF00340
 ERF00350
 ERF00360
 ERF00370
 ERF00380
 ERF00390
 ERF00400
 ERF00410
 ERF00420
 ERF00430
 ERF00440
 ERF00450
 ERF00460
 ERF00470
 ERF00480
 ERF00490
 ERF00500
 ERF00510
 ERF00520
 ERF00530
 ERF00540
 ERF00550
 ERF00560
 ERF00570
 ERF00580

```

SUBROUTINE FIT(X,F,A,B,C)          FIT00190
CQUADRATIC FIT TO THREE POINTS USING NEWTON'S FUNDAMENTAL FORMULA
INPUTS                                     FIT00010
X   - 3 VALUES OF THE INDEPENDENT VARIABLE    FIT00020
F   - 3 FUNCTION VALUES CORRESPONDING TO THE X VALUES    FIT00030
OUTPUTS                                     FIT00040
A   - COEFFICIENT OF THE X**2 TERM           FIT00050
B   - COEFFICIENT OF THE X TERM               FIT00060
C   - CONSTANT TERM                         FIT00070
*****                                     FIT00080
DIMENSION X(3),F(3)                      FIT00090
H=X(2)-X(1)                                FIT00100
DF1=(F(2)-F(1))/H                          FIT00110
DF2=(F(3)-2.*F(2)+F(1))/(2.*H**2)        FIT00120
A=DF2                                       FIT00130
B=DF1-DF2*(X(2)+X(1))                     FIT00140
C=F(1)+X(1)*(X(2)*DF2-DF1)                 FIT00150
RETURN                                     FIT00160
END                                         FIT00170
FIT00180
FIT00200
FIT00210
FIT00220
FIT00230
FIT00240
FIT00250
FIT00260
FIT00270
FIT00280

```

```

FUNCTION FUNCT(X,Z) FUC00010
LOGICAL HORIZ,SKIP FUC00020
COMMON /CLOCK/ TIME,TWIND FUC00030
COMMON/MODE/HORIZ FUC00040
COMMON/SKIPIT/SKIP FUC00050
*****
C *****
PURPOSE FUC00060
TO SUPPLY A TRANSMITTANCE FUNCTION FOR THE CONTOUR TRACING
ROUTINE IN ORDER TO DETERMINE THE CLOUD EDGE. FUC00070
FUC00080
FUC00090
FUC00100
FUC00110
FUC00120
FUC00130
FUC00140
FUC00150
FUC00160
FUC00170
FUC00180
FUC00190
FUC00200
FUC00210
FUC00220
FUC00230
FUC00240
FUC00250
FUC00260
FUC00270
FUC00280
FUC00290
FUC00300
FUC00310
FUC00320
FUC00330
FUC00340
FUC00350
FUC00360
FUC00370
FUC00380
FUC00390
FUC00400
FUC00410
FUC00420
FUC00430
FUC00440
FUC00450
FUC00460
FUC00470
INPUT
X THE HORIZONTAL COORDINATE IN METERS
Z THE VERTICAL COORDINATE IN METERS
OUTPUT
RETURNS THE LOG OF THE OPTICALLY WEIGHTED CL VALUE (AT
VISIBLE WAVELENGTHS) FOR THE LINE OF SIGHT SPECIFIED BY X,Z
FUNCTIONS CALLED
CWIND
CALLED BY GFUN, CLIMB, GRAD2
*****
HORIZ=.TRUE.
SKIP=.TRUE.
Y=0.
EXT1=0.0
EXT2=0.0
IF(Z.LE.0.)GO TO 100
EXT1=CWIND(X,Y,Z,TIME)
IF(TIME.LE.TWIND)GO TO 10
EXT2=CSPHER(X,Y,Z,TIME)
10 EXT=EXT1+EXT2
IF(EXT.LE.1.E-30)GO TO 100
FUNCT= ALOG(EXT)
GO TO 999
100 FUNCT=-30.
999 CONTINUE
RETURN
END

```

```

SUBROUTINE GAMMA(XX,GX,IER)
SUBROUTINE GAMMA

PURPOSE
    COMPUTES THE GAMMA FUNCTION FOR A GIVEN ARGUMENT

USAGE
    CALL GAMMA(XX,GX,IER)

DESCRIPTION OF PARAMETERS
    XX -THE ARGUMENT FOR THE GAMMA FUNCTION
    GX -THE RESULTANT GAMMA FUNCTION
    IER -THE RESULTANT ERROR CODE WHERE
        IER=0 NO ERROR
        IER=1 XX IS WITHIN .000001 OF BEING A NEGATIVE INTEGER
        IER=2 XX GT 57, OVERFLOW, GX SET TO 1.E32

COMMENTS
    NONE

SUBROUTINES AND FUNCTIONS
    NONE

METHOD
    THE RECURSION RELATION AND POLYNOMIAL APPROXIMATION
    BY C. HASTINGS, JR. 'APPROXIMATIONS FOR DIGITAL COMPUTERS',
    PRINCETON UNIVERSITY PRESS, 1955
.....
4 IF(XX-57.) 6,6,4
4 IER=2
5 GX=1.E32
6 RETURN
6 X=XX
7 ERR=1.0E-6
8 IER=0
9 GX=1.0
10 IF(X<-2.0) 50,50,15
10 X=X-1.0
11 GX=GX*X
12 GO TO 10
50 IF(X<-1.0) 60,120,110
      SEE IF X IS NEAR NEGATIVE INTEGER OR ZERO
60 IF(X-ERR) 62,62,80
62 Y=FLOAT(INT(X))-X
63 IF(ABS(Y)-ERR) 130,130,70
      X NOT NEAR A NEGATIVE INTEGER OR ZERO
70 IF(X-1.0)>0,80,110
80 GX=GX/X
81 X=X+1.0
82 GO TO 70
110 Y=X-1.0
     GY=1.0+Y*(-0.5771017+Y*(0.9858540+Y*(-0.8764218+Y*(0.8328212+
     1*Y*(-0.5684729+Y*(0.2548205+Y*(-0.05149930)))))))>
     GX=GX*GY
120 RETURN
130 IER=1
131 RETURN
END

```

167

FUNCTION GFUN(S) GFU00010
C GFU00020
CCC GFU00030
C GFU00040
C GFU00050
C GFU00060
C GFU00070
C COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON GFU00080
CALL VSUM(BASE,DIR,S,P) GFU00090
GFUN=FUNCT(P(1),P(2)) GFU00100
RETURN GFU00110
END GFU00120

SUBROUTINE GRAD2(PT,FUNCT,RES,GRAD,SLOPE)

THIS SUBROUTINE CALCULATES THE UNIT GRADIENT VECTOR
OF A FUNCTION AT THE GIVEN POINT USING THE FORMULA
PARTIAL DF/DX = $(F(X+R,Y) - F(X,Y))/R$ WHERE R IS
SMALL, SIMILARLY FOR PARTIAL DF/DY. IT THEN
NORMALIZES THE RESULTANT VECTOR FOR THE UNIT GRADIENT
AND FINDS THE SLOPE, WHICH IS THE MAGNITUDE OF
OF THE REGULAR GRADIENT.

ARGUMENTS PASSED

PT-THE POINT AT WHICH THE UNIT GRADIENT IS FOUND.
FUNCT-THE FUNCTION(X,Y)
RES-R
GRAD-THE UNIT GRADIENT
SLOPE-SLOPE AT PT

OTHER VARIABLES

C00,C10,C01-THE FUNCTION AT THE POINTS F(X,Y),F(X+R,Y),
AND F(X,Y+R) RESPECTIVELY.

SUBROUTINES CALLED

UNIT-NORMALIZES A VECTOR AND FINDS ITS MAGNITUDE.

```
DIMENSION PT(2),GRAD(2)
C00=FUNCT(PT(1),PT(2))
C10=FUNCT(PT(1)+RES,PT(2))
C01=FUNCT(PT(1),PT(2)+RES)
GRAD(1)=(C10-C00)/RES
GRAD(2)=(C01-C00)/RES
CALL UNIT(GRAD,GRAD,SLOPE)
RETURN
END
```

GRA00240
GRA00010
GRA00020
GRA00030
GRA00040
GRA00050
GRA00060
GRA00070
GRA00080
GRA00090
GRA00100
GRA00110
GRA00120
GRA00130
GRA00140
GRA00150
GRA00160
GRA00170
GRA00180
GRA00190
GRA00200
GRA00210
GRA00220
GRA00230
GRA00240
GRA00250
GRA00260
GRA00270
GRA00280
GRA00290
GRA00300
GRA00310
GRA00320
GRA00330

```

SUBROUTINE GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)           GRD00320
ROUTINE TO EVALUTE THE INTEGRAND FOR THE TRAPEZOIDAL INTEGRATION   GRD00010
FOR FINDING THE OPTICALLY WEIGHTED CONCENTRATION ALONG THE LINE OF   GRD00020
SIGHT WHERE THE DUST IS GENERATED BY A VEHICLE.                  GRD00030
GRD00040
GRD00050
GRD00060
GRD00070
GRD00080
GRD00090
GRD00100
GRD00110
GRD00120
GRD00130
GRD00140
GRD00150
GRD00160
GRD00170
GRD00180
GRD00190
GRD00200
GRD00210
GRD00220
GRD00230
GRD00240
GRD00250
GRD00260
GRD00270
GRD00280
GRD00290
GRD00300
GRD00310
GRD00320
GRD00330
GRD00340
GRD00350
GRD00360
GRD00370
GRD00380
GRD00390
GRD00400
GRD00410
GRD00420
GRD00430
GRD00440
GRD00450
GRD00460
GRD00470
GRD00480
GRD00490
GRD00500
GRD00510
GRD00520
GRD00530
GRD00540
GRD00550
GRD00560
GRD00570
GRD00580
GRD00590
GRD00600
GRD00610
GRD00620
GRD00630
GRD00640
GRD00650
GRD00660
GRD00670
GRD00680
GRD00690
GRD00700

C INPUTS
U      - A UNIT VECTOR ALONG THE LINE OF SIGHT               GRD00010
TR     - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE SYSTEM   GRD00020
XNORM - DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER             GRD00030
TIME   - PRESENT TIME AT WHICH A TRANSMITTANCE IS WANTED          GRD00040
TIVEN- TIME THAT THE VEHICLE HAS TRAVELED                      GRD00050
VDIR   - VECTOR CONTAINING DESCRIBING THE VEHICLE DIRECTION AND SPEED   GRD00060
GRD00070
GRD00080
GRD00090
GRD00100
GRD00110
GRD00120
GRD00130
GRD00140
GRD00150
GRD00160
GRD00170
GRD00180
GRD00190
GRD00200
GRD00210
GRD00220
GRD00230
GRD00240
GRD00250
GRD00260
GRD00270
GRD00280
GRD00290
GRD00300
GRD00310
GRD00320
GRD00330
GRD00340
GRD00350
GRD00360
GRD00370
GRD00380
GRD00390
GRD00400
GRD00410
GRD00420
GRD00430
GRD00440
GRD00450
GRD00460
GRD00470
GRD00480
GRD00490
GRD00500
GRD00510
GRD00520
GRD00530
GRD00540
GRD00550
GRD00560
GRD00570
GRD00580
GRD00590
GRD00600
GRD00610
GRD00620
GRD00630
GRD00640
GRD00650
GRD00660
GRD00670
GRD00680
GRD00690
GRD00700

C OUTPUT
VALUE - VALUE OF THE INTEGRAND                               GRD00010
FUNCTIONS AND SUBROUTINES NEEDED
CONLEN - TO FIND THE LENGTH OF THE INTERSECTION OF THE LINE OF SIGHT AND THE TILTED CYLINDER   GRD00020
GRD00030
GRD00040
GRD00050
GRD00060
GRD00070
GRD00080
GRD00090
GRD00100
GRD00110
GRD00120
GRD00130
GRD00140
GRD00150
GRD00160
GRD00170
GRD00180
GRD00190
GRD00200
GRD00210
GRD00220
GRD00230
GRD00240
GRD00250
GRD00260
GRD00270
GRD00280
GRD00290
GRD00300
GRD00310
GRD00320
GRD00330
GRD00340
GRD00350
GRD00360
GRD00370
GRD00380
GRD00390
GRD00400
GRD00410
GRD00420
GRD00430
GRD00440
GRD00450
GRD00460
GRD00470
GRD00480
GRD00490
GRD00500
GRD00510
GRD00520
GRD00530
GRD00540
GRD00550
GRD00560
GRD00570
GRD00580
GRD00590
GRD00600
GRD00610
GRD00620
GRD00630
GRD00640
GRD00650
GRD00660
GRD00670
GRD00680
GRD00690
GRD00700

C ****
DIMENSION U(3),TR(3),VDIR(2),VP(2)                         GRD00010
COMMON/M05/DMMY(604),DMYC(600),
+          ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
+          RB(3,25),Z2(3,25)
COMMON /I0UNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUGRD00020
COMMON/VL/VLOAD
COMMON/CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDECK
C FIND THE VEHICLE POSITION AT TIME TIVEN
C
VP(1)=TIVEN*VDIR(1)
VP(2)=TIVEN*VDIR(2)
TOF=TIME-TIVEN
DO 10 I=1,ICOUNT
IND=I
IF(TOF.LT.TIMES(I))GO TO 20
10 CONTINUE
C
IF TOF (TIME OF FLIGHT) IS GREATER THAN TABULATED VALUES IT IS
ASSUMED THE THE CLOUD HAS DISSIPATED
XC=X1*Z+X0 IS THE LINE THRIUGH THE CENTER OF THE CYLINDER
C
GO TO 50
20 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)
X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)
RAD=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND)
HTTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)
HTBOT=0.0
RTOP=RAD
RBOT=RAD
XCEN=VP(1)+(X1*HTTOP+X0)
YCEH=VP(2)
XB=VP(1)+(X1*HTBOT+X0)
YB=VP(2)
IF(U(3).LT.1.E-06)GO TO 30
C COMPUTE INTERSECTION LENGTH FOR NON HORIZONTAL LINES OF SIGHT

```

```

CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,XNORM,PLEN)GRD00710
GO TO 40 GRD00720
GRD00730
CC DETERMINE LENGTH OF INTERSECTION FOR A HORIZONTAL LINE OF SIGHT GRD00740
30 IF(HTTOP.LT.TR<3>)GO TO 35 GRD00750
A=U<1>**2+U<2>**2 GRD00760
B=U<1>*(TR<1>-XCEN)+U<2>*(TR<2>-YCEN) GRD00770
C=(TR<1>-XCEN)**2+(TR<2>-YCEN)**2-RAD**2 GRD00780
X=B**2-A*C GRD00790
IF(X.LT.0.0)GO TO 35 GRD00800
P1=(-B+SQRT(X))/A GRD00810
P2=(-B-SQRT(X))/A GRD00820
IF(P1.GT.XNORM.AND.P2.GT.XNORM)GO TO 35 GRD00830
IF(P2.LT.0.0.AND.P1.LT.0.0)GO TO 35 GRD00840
PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0) GRD00850
GO TO 40 GRD00860
35 PLEN=0.0 GRD00870
40 VOL=PI*HTTOP*(RAD**2) GRD00880
VALUE=VLOAD*PLEN/VOL GRD00890
GO TO 999 GRD00900
50 VALUE=0.0 GRD00910
999 RETURN GRD00920
END GRD00930
GRD00940

```

```

SUBROUTINE GREEN(Z,Z1,T,ALPHA,T0,IER)
*****  

C PURPOSE  

TO COMPUTE THE GENERALIZED GREENS FUNCTION  

USES GREEN1  

SEE GREEN1 FOR ARGUMENT LIST  

*****  

REAL N,M
COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
C IF(N .EQ. 1.) GO TO 2
X2=2.-N
AT=ALPHA*T
T0=0.
IF(AT .GE. Z1) RETURN
CALL GREEN1((Z+AT)**X2,Z1**X2,X2*X2*T,(N-1.)/X2,T1,IER)
T1=T1*X2*Z1**(1.-N)
U=1,
T2=0,
IF(ABS(ALPHA) .LT. 1.E-4) GO TO 1
ZMZ=Z-Z1+AT
X2=N+1.
AN1=ALPHA*X2
ZMZN=Z1**X2 - (Z1-AT)**X2
ARG=(-AN1*ZMZ*ZMZ)/(4.*ZMZN)
IF(ARG .LT. -70.) GO TO 3
T2=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
3 IF(T1.LT.1.E-30 .AND. T2.LT.1.E-30) RETURN
C CALCULATION OF MIXING RATIO, U, BY N=1 ANALOGY
CALL GREEN1(Z+AT,Z1,T,0.,T1U,IER)
X2=2,
AN1=ALPHA*X2
ZMZN=Z1**X2 - (Z1-AT)**X2
T2U=0.
ARG=(-AN1*ZMZ*ZMZ)/(4.*ZMZN)
IF(ARG .LT. -70.) GO TO 4
T2U=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
4 IF(T1U.LT.1.E-30 .AND. T2U.LT.1.E-30) GO TO 1
CALL GREEN1(Z,Z1,T,ALPHA,G,IER)
U=(G-T2U)/(T1U-T2U)
1 IF(U .LT. 0.) U=0.
IF(U .GT. 1.) U=1.
C COMBINE LIMITING SOLUTIONS WITH DETERMINED MIXING RATIO
T0=U*T1 + (1.-U)*T2
RETURN
2 CALL GREEN1(Z,Z1,T,ALPHA,T0,IER)
RETURN
END

```

GRE00010
GRE00020
GRE00030
GRE00040
GRE00050
GRE00060
GRE00070
GRE00080
GRE00090
GRE00100
GRE00110
GRE00120
GRE00130
GRE00140
GRE00150
GRE00160
GRE00170
GRE00180
GRE00190
GRE00200
GRE00210
GRE00220
GRE00230
GRE00240
GRE00250
GRE00260
GRE00270
GRE00280
GRE00290
GRE00300
GRE00310
GRE00320
GRE00330
GRE00340
GRE00350
GRE00360
GRE00370
GRE00380
GRE00390
GRE00400
GRE00410
GRE00420
GRE00430
GRE00440
GRE00450
GRE00460
GRE00470
GRE00480
GRE00490
GRE00500
GRE00510
GRE00520
GRE00530
GRE00540
GRE00550
GRE00560
GRE00570
GRE00580
GRE00590

```

SUBROUTINE GREEN1(Z,Z1,T,NU,BI,IER) GRV00490
  SUBROUTINE GREEN1 GRV00010
    PURPOSE GRV00020
      COMPUTE THE I BESSSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER GRV00030
      AND MULTIPLY BY AN APPROPRIATE POWER OF THE ARGUMENT GRV00040
      AND AN EXPONENTIAL IN ORDER TO CALCULATE THE GREENS GRV00050
      FUNCTION FOR THE WIND DIFFUSION EQUATION GRV00060
    USAGE GRV00070
      CALL GREEN1(Z,Z1,T,NU,BI,IER) GRV00080
    DESCRIPTION OF PARAMETERS GRV00090
      Z,Z1,T -THE ARGUMENTS OF THE FUNCTION DESIRED GRV00100
      NU -THE ORDER OF THE I BESSSEL FUNCTION GRV00110
      BI -THE RESULTANT BESSSEL FUNCTION GRV00120
      IER -RESULTANT ERROR CODE WHERE GRV00130
        IER=-1 EXPONENTIAL UNDERFLOW (NON-FATAL), BI SET TO 0.0 GRV00140
        IER=0 NO ERROR GRV00150
        IER=1 NU NEAR NEGATIVE INTEGER GRV00160
        IER=2 OVERFLOW IN GAMMA GRV00170
        IER=3 UNDERFLOW, BI .LT. 1.E-32, BI SET TO 0.0 GRV00180
        IER=4 OVERFLOW, X .GT. 90 WHERE X .GT. N GRV00190
        IER=5 X IS NEGATIVE GRV00200
    REMARKS GRV00210
      NU IS A REAL NUMBER GRV00220
      N AND X MUST BE .GE. ZERO GRV00230
      THIS SUBROUTINE IS A MODIFICATION OF BESI WHICH COMPUTES THE GRV00240
      I BESSSEL FUNCTION FOR INTEGER ORDERS. THE CHANGE REQUIRES GRV00250
      USE OF THE GAMMA FUNCTION FOR COMPUTING THE FIRST TERM OF THE GRV00260
      SERIES. THE SUCCESSIVE TERMS ARE CALCULATED WITH THE SAME GRV00270
      RECURSION FORMULA AND THE ASYMPTOTIC APPROXIMATION IS ALSO GRV00280
      UNCHANGED. BESI IS IN THE IBM SYSTEM/360 SCIENTIFIC GRV00290
      SUBROUTINE PACKAGE. MODIFICATIONS MADE BY D. DVURE, AERODYNEGRV00300
      RESEARCH INC. JANUARY 15, 1979. GRV00310
    SUBROUTINES AND FUNCTIONS REQUIRED GRV00320
      GAMMA WHICH COMPUTES THE GAMMA FUNCTION GRV00330
    METHOD GRV00340
      COMPUTES I BESSSEL FUNCTION USING SERIES OR ASYMPTOTIC GRV00350
      APPROXIMATION DEPENDING ON THE RANGE OF THE ARGUMENT. GRV00360
    CALLED BY MOMENT GRV00370
    .....
    REAL NU GRV00380
    X=2.*SQRT(Z*Z1)/T GRV00390
    CHECK FOR ERRORS IN NU AND X AND EXIT IF ANY ARE PRESENT GRV00400
    ..... GRV00410
    IER=0 GRV00420
    BI=1.0 GRV00430
    IFC(NU>10,15,10 GRV00440
    10 IFC(X>160,20,20 GRV00450
    15 IFC(X>160,17,20 GRV00460
    17 ARG=-(Z+Z1)/T GRV00470
    IFC(ARG .LT. -80.) GO TO 170 GRV00480
    BI=EXP(ARG)/T GRV00490
    RETURN GRV00500
    DEFINE TOLERANCE GRV00510
    20 TOL=1.E-3 GRV00520
    IF ARGUMENT GT 12 AND GT NU, USE ASYMPTOTIC FORM GRV00530
    ..... GRV00540
    ..... GRV00550
    ..... GRV00560
    ..... GRV00570
    ..... GRV00580
    ..... GRV00590
    ..... GRV00600
    ..... GRV00610
    ..... GRV00620
    ..... GRV00630
    ..... GRV00640
    ..... GRV00650
    ..... GRV00660
    ..... GRV00670
    ..... GRV00680
    ..... GRV00690
    ..... GRV00700

```

```

      IF(X-12.)40,40,30          GRV00710
30 IF(X-ABS(NU))40,40,110      GRV00720
C COMPUTE FIRST TERM OF SERIES AND SET INITIAL VALUE OF THE SUM   GRV00730
C
40 XX=X/2,                      GRV00740
N=INT(NU)                      GRV00750
FN=N                          GRV00760
R=NU-FN                        GRV00780
CALL GAMMA(1.+NU,GR,IER)        GRV00800
IF(IER.EQ.0) GO TO 60           GRV00810
50 BI=0.0                         GRV00820
RETURN                         GRV00830
60 TERM=1./GR                     GRV00840
70 BI=TERM                        GRV00850
XX=XX*XX                        GRV00860
C COMPUTE TERMS, STOPPING WHEN ABS(TERM) LE ABS(SUM OF TERMS)*TOLERAGRV00870
C
DO 90 K=1,1000                  GRV00880
IF(ABS(TERM)-ABS(BI*TOL))95,95,80      GRV00890
80 FK=K                          GRV00910
FK=FK*(NU+FK)                   GRV00920
TERM=TERM*(XX/FK)                GRV00930
90 BI=BI+TERM                   GRV00940
95 ARG=-(Z+21)/T                 GRV00950
IF(ARG.LT.-80.) GO TO 170         GRV00970
BI=BI*(Z1/T)**NU*EXP(ARG)/T       GRV00980
C RETURN BI AS ANSWER            GRV01000
C
100 RETURN                       GRV01010
C X GT 12 AND X GT NU, SO USE ASYMPTOTIC APPROXIMATION          GRV01020
C
110 FN=4.*NU*NU                  GRV01030
115 XX=1./(8.*X)                 GRV01040
TERM=1.                         GRV01050
BI=1.                           GRV01060
DO 130 K=1,30                   GRV01070
IF(ABS(TERM)-ABS(BI*TOL)) 140,140,120      GRV01080
120 FK=(2*K-1)**2               GRV01090
TERM=TERM*XX*(FK-FN)/FLOAT(K)        GRV01100
130 BI=BI+TERM                  GRV01110
C SIGNIFICANCE LOST AFTER 30 TERMS, TRY SERIES                  GRV01120
C
GO TO 40                         GRV01130
140 PI=3.141592653               GRV01140
ARG=X-(Z+21)/T                   GRV01150
IF(ARG.LT.-80.) GO TO 170         GRV01160
BI=BI*(Z1/Z)**(NU/2.)*EXP(ARG)/SQRT(2.*PI*X)/T      GRV01170
GO TO 100                         GRV01180
160 IER=5                         GRV01190
GO TO 100                         GRV01200
170 BI=0.0                         GRV01210
GO TO 50                          GRV01220
END                               GRV01230
                                         GRV01240
                                         GRV01250
                                         GRV01260
                                         GRV01270
                                         GRV01280

```

```

SUBROUTINE MOMENT(VGRAV,ZIN,H,TIN,Q,XBAR,SIGW2,SIGP2)      MOM00010
  REAL M,N,NM                                                 MOM00020
  DIMENSION AL(9),Z(9),T(9),XB(81,4,3),SW(81,4,3),SP(81,4,3),NM(9)   MOM00030
  DIMENSION VAL(16),XVAL(8),W(8),XI(4),IB(4),NTC(4),II(4),X(9,4)    MOM00040
  LOGICAL FIRST                                              MOM00050
  COMMON /WNDPRM/DXZ0,DYX0,D20,U0,M,N,ZINV                  MOM00060
  COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNT,NDIRTU,NCLIMT,KSTOR,NPLOTUMOM00070
  EQUIVALENCE (Z(1),X(1,1)),(T(1),X(1,2)),(AL(1),X(1,3))        MOM00080
  EQUIVALENCE (NM(1),X(1,4))
  DATA FIRST/.TRUE./,IB/4,4,1,2/,ITC/11/,HREF/1./             MOM00090
C *****
C PURPOSE
C
C TO CONVERT PARAMETERS TO NONDIMENSIONAL FORM AND THEN COMPUTE
C THE ZERO ORDER MOMENT AND INTERPOLATE FROM TABULATED VALUES OF
C THE HIGHER ORDER MOMENTS
C
C INPUT
C
C VGRAV   THE GRAVITATIONAL SETTLING VELOCITIES OF THE PARTICLE
C           IN METERS / SEC
C ZIN      THE HEIGHT < METERS > AT WHICH THE MOMENTS ARE DESIRED
C H        THE HEIGHT OF RELEASE OF THE PARTICLES IN METERS
C TIN      THE TIME IN SECONDS AFTER RELEASE
C
C OUTPUT
C
C Q        THE VERTICAL CONCENTRATION OF PARTICLES AT HEIGHT Z
C XBAR     THE DISPLACEMENT < METERS > IN THE X (IE WIND) DIRECTION
C           OF THE CENTER OF MASS OF PARTICLES AT HEIGHT Z
C SIGW2    THE SQUARE OF THE STANDARD DEVIATION OF THE WINDWARD
C           DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
C SIGP2    THE SQUARE OF THE STANDARD DEVIATION OF THE CROSS-WIND
C           DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
C
C SUBROUTINES CALLED
C
C DTERPS   PUTS THE NEEDED VALUES OF THE TABULATED MOMENTS
C           INTO A ONE-DIMENSIONAL ARRAY
C DTERPI   A FUNCTION WHICH RETURNS THE INTERPOLATED VALUE
C           FOR GIVEN ARGUMENTS AND ARRAYS
C GREEN    CALCULATES THE GREENS FUNCTION WHICH IS THE
C           0-ORDER MOMENT
C
C CALLED BY CWIND
C *****
C IF(.NOT.FIRST)GO TO 5
C READ IN THE TABLE OF MOMENTS ON THE FIRST CALL OF MOMENT
C
C Z        LOG OF NON-DIMENSIONAL HEIGHTS AT WHICH MOMENTS ARE TABULATED
C T        LOG OF NON-DIMENSIONAL TIMES AT WHICH MOMENTS ARE TABULATED
C AL       NON-DIMENSIONAL SETTLING VELOCITIES AT WHICH MOMENTS ARE
C           TABULATED
C NM       DIFFUSIVITY POWER LAW EXPONENTS AT WHICH MOMENTS ARE
C           TABULATED
C XB       TABULATED VALUES OF LOGS OF FIRST ORDER MOMENTS (RELATED
C           TO MEAN HORIZONTAL DISPLACEMENT)
C SW       TABULATED VALUES OF LOGS OF WIND SHEAR COMPONENT OF SECOND
C           ORDER MOMENT (CONTRIBUTES TO VARIANCE IN WIND DIRECTION)
C SP       TABULATED VALUES OF LOGS OF SECOND ORDER MOMENT COMMON TO
C           WIND AND CROSS-WIND VARIANCES
C
C READ(NDIRTU,1) NZ,NT,NA,NM
C 1 FORMAT(413)

```

```

      NTC(1)=NZ-1          M0M00710
      NTC(2)=NT-1          M0M00720
      NTC(3)=NA-1          M0M00730
      NTC(4)=NN-1          M0M00740
      READ(NDIR TU,2) (Z(I),I=1,NZ)          M0M00750
      2 FORMAT(6E13.5)        M0M00760
      READ(NDIR TU,2) (T(I),I=1,NT)          M0M00770
      READ(NDIR TU,2) (AL(I),I=1,NA)          M0M00780
      READ(NDIR TU,2) (NM(I),I=1,NN)          M0M00790
      NZT=NZ*NT             M0M00800
      DO 3 L=1,NN            M0M00810
      READ(NDIR TU,2) ((XB(IJ,K,L),IJ=1,NZT),K=1,NA)    M0M00820
      READ(NDIR TU,2) ((SW(IJ,K,L),IJ=1,NZT),K=1,NA)    M0M00830
      READ(NDIR TU,2) ((SP(IJ,K,L),IJ=1,NZT),K=1,NA)    M0M00840
      3 CONTINUE             M0M00850
      FIRST=.FALSE.          M0M00860
      REWIND NDIR TU         M0M00870
      5 CONTINUE             M0M00880
C
C     CONVERT INPUT PARAMETERS TO NONDIMENSIONAL FORM           M0M00890
C
      SCLU=DZ0+H**(N-1.)          M0M00900
      XI(1)=ZIN/H             M0M00910
      XI(2)=SCLU*TIN/H          M0M00930
      XI(3)=VGRAV/SCLU          M0M00940
      XI(4)=N                  M0M00950
      CALL GREEN(XI(1),HREF,XI(2),XI(3),Q,IER)          M0M00960
      Q=Q/H                  M0M00970
      IF(Q.LE. 1.E-10) GO TO 999          M0M00980
C
C     TAKE LOGS FOR LOGARITHMIC INTERPOLATION           M0M01000
C
      XI(1)= ALOG(XI(1))          M0M01010
      XI(2)= ALOG(XI(2))          M0M01020
C
C     DETERMINE INDICES OF LOWEST CORNER POINT OF THE CUBE TO   M0M01030
C     BE USED IN INTERPOLATION MAKING SURE THAT ENOUGH CORNER POINTS   M0M01040
C     OF THE CUBE HAVE TABULATED VALUES                      M0M01050
C
      DO 100 I=1,4          M0M01060
      II(I)=IB(I)          M0M01070
      100 CONTINUE          M0M01080
      DO 101 III=1,4          M0M01090
      I=5-III          M0M01100
      6 IA=II(I)          M0M01110
      IF(XI(I).GE. X(IA,I).AND. XI(I).LE. X(IA+1,I)) GO TO 101    M0M01120
      IF(XI(I).LT. X(IA,I).AND. IA.EQ. 1) GO TO 101    M0M01130
      IF(XI(I).GT. X(IA,I).AND. IA.EQ. NTC(I)) GO TO 101    M0M01140
      ISAY=II(I)
      II(I)=IA + IFIX(SIGN(1.,XI(I)-X(IA,I)))
      IT=0          M0M01150
      DO 102 JI=1,2          M0M01160
      JIX=JI + II(1) - 1          M0M01170
      DO 102 IJ=1,2          M0M01180
      IJX=JIX + (IJ + II(2) - 2)*NZ          M0M01190
      DO 102 K=1,2          M0M01200
      KX=K-1 + II(3)          M0M01210
      DO 102 L=1,2          M0M01220
      LX=L-1 + II(4)          M0M01230
      102 IF(XB(IJX,KX,LX).GT. -100.) IT=IT+1          M0M01240
      CONTINUE          M0M01250
      IF(IT.GT. IT0) GO TO 6          M0M01260
      101 II(I)=ISAY          M0M01270
      CONTINUE          M0M01280
C
C     PERFORM THE INTERPOLATION WITH DETERMINED CUBE OF POINTS   M0M01290
C
      DO 103 I=1,4          M0M01300
      I2=I*2          M0M01310
      II=I2-1          M0M01320

```

```

IA=II(I)
XVAL(I1)=X(IA,I)
XVAL(I2)=X(IA+1,I)
103 CONTINUE
CALL DTERPS(II,XB,VAL,NZ)                               MOM01410
XBAR=DTERPI(-4,XI,XVAL,VAL,-100.,W)                     MOM01420
CALL DTERPS(II,SW,VAL,NZ)                               MOM01430
SIGW2=DTERPI(-4,XI,XVAL,VAL,-100.,W)                     MOM01440
CALL DTERPS(II,SP,VAL,NZ)                               MOM01450
SIGP2=DTERPI(-4,XI,XVAL,VAL,-100.,W)                     MOM01460
CCC CONVERT THE LOG OF THE NONDIMENSIONAL VALUES INTERPOLATED    MOM01470
TO THE USUAL DIMENSIONAL FORM                                MOM01480
C
SCL=U0*H**((M+1,))/SCLU                                 MOM01490
XBAR=SCL*EXP(XBAR)                                     MOM01500
SIGW2=SCL*SCL*EXP(SIGW2)                               MOM01510
SIGP2=2.*DX20*H*H*EXP(SIGP2)                           MOM01520
SIGW2=SIGW2+SIGP2                                      MOM01530
SIGP2=DYX0*SIGP2                                       MOM01540
999 RETURN                                              MOM01550
END                                                    MOM01560

```

SUBROUTINE PATH(T,U,XCEN,YCEN,RAD,PLEN) PATH0290
 THIS SUBROUTINE COMPUTES THE PATH LENGTH THROUGH THE SPHERE OR PATH0010
 WAKE FOR A HORIZONTAL PATH. PATH0020
 INPUTS PATH0030
 T - TRANSMITTER COORDINATE IN THE LOCAL COORDINATE SYSTEM PATH0040
 U - UNIT VECTOR ALONG THE LINE CONNECTING THE TRANSMITTER PATH0050
 AND RECEIVER PATH0060
 XCEN - X COORDINATE OF THE CENTER OF THE CIRCLE PATH0070
 YCEN - Y COORDINATE OF THE CENTER OF THE CIRCLE PATH0080
 RAD - RADIUS AT THE DESIRED HEIGHT PATH0090
 OUTPUT PATH0100
 PLEN - LENGTH OF THE INTERSECTION OF THE CONE AT HEIGHT T(3) PATH0110
 AND THE LINE OF SIGHT PATH0120
 FUNCTIONS AND SUBROUTINES NEEDED PATH0130
 NONE PATH0140
***** PATH0150
 DIMENSION T(3),U(3) PATH0160
 A=U(1)**2+U(2)**2 PATH0170
 PLEN=0, PATH0180
 X=RAD**2*A-(U(2)*(T(1)-XCEN)-U(1)*(T(2)-YCEN))**2 PATH0190
 IF (X.GT.0.) PLEN=2.*SQRT(X)/A PATH0200
 RETURN PATH0210
 END PATH0220
***** PATH0230
***** PATH0240
***** PATH0250
***** PATH0260
***** PATH0270
***** PATH0280
***** PATH0290
***** PATH0300
***** PATH0310
***** PATH0400

```
SUBROUTINE PERP(A,B)
DIMENSION A(2),B(2)
C *** B IS ROTATED 90 DEGREES COUNTERCLOCKWISE FROM A
B(1)=-A(2)
B(2)=A(1)
RETURN
END
```

```
PER00010
PER00020
PER00030
PER00040
PER00050
PER00060
PER00070
```

SUBROUTINE PRECL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,PHI,DHDT,
1 CHWT,NCHRG,DETDEP,HSOIL,DSOD,SILT) PRE00350
PRE00360
PRE00010

ROUTINE TO PRECOMPUTE EXPLOSION PRODUCED DUST CLOUD AND STORE ON
EXTERNAL FILE UNIT IFILE PRE00020
PRE00030
PRE00040

OUTPUTS PRE00050
PRE00060
PRE00070

XC1 COEFFICIENTS OF QUADRATIC FIT TO SLOPE OF LINE DESCRIBING
X DISPLACEMENT OF CONE (ALONGWIND) PRE00080
PRE00090

XCO COEFFICIENTS OF QUADRATIC FIT TO CONSTANT TERM OF LINE
DESCRIBING X DISPLACEMENT OF CONE (ALONGWIND) PRE00100
PRE00110

Z2 COEFFICIENTS OF QUADRATIC FIT TO HEIGHT OF TOP OF CLOUD PRE00120
PRE00130

RT COEFFICIENTS OF QUADRATIC FIT TO THE RADIUS OF THE TOP OF THE
CONE PRE00140
PRE00150

RB COEFFICIENTS OF THE QUADRATIC FIT TO THE RADIUS OF THE CONE
AT A HEIGHT OF THE AVERAGE OF THE DISC SOURCES PRE00160
PRE00170

THE ABOVE OUTPUT ARE THE COEFFICIENTS OF QUADRATIC FITS THROUGH
THREE CONSECUTIVE POINT IN TIME. THE QUADRATIC FITS ARE STORED IN
COMMON/ M05 / WITH THE ARRAY TIMES CONTAINING THE LAST TIME
OF EACH QUADRATIC PIECE WITH THE FIRST STARTING AT 0.0
THE FITS ARE WRITTEN ONTO A FILE INDICATED BY IFILE USING
A BINARY WRITE
THE FITS ARE STORED SUCH THAT

F(TIME)=VAR(1,J)*TIME**2 + VAR(2,J)*TIME + VAR(3,J) PRE00220
PRE00230

AND TIMES(J-1) < TIME < TIMES(J) PRE00240
PRE00250

***** LOGICAL SWITCH,CHANGE,DHDT PRE00260
PRE00270

REAL KZ,KX PRE00380

DIMENSIÓN T(3),FRB(3),FRT(3),FXC1(3),FXC0(3),XB(3),OWF(3,2) PRE00390

DIMENSION F22(3),ZTEMP(2),TMPMES(2),ZWND(2),WNDMES(2),OWFC(5) PRE00400

COMMON /I0UNIT/I0IN,I0OUT,IPHUFN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUP,PRE00410

COMMON/OPTION/I0PT,IFILE PRE00420

COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20,3)PRE00430

COMMON/BUDYCL/RSPH,DELT,YZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM PRE00440

COMMON/M05/DMMMY(604),DMM(600) PRE00450

+ ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25), PRE00460

+ RB(3,25),Z2(3,25) PRE00470

COMMON/VL/VLOAD PRE00480

COMMON/SIG/SIG02,SIGC PRE00490

COMMON/CLOCK/FTIME,WIND PRE00500

COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE PRE00510

COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV PRE00520

COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2) PRE00530

COMMON/RCARB/RCARB1,RCARB2 PRE00540

COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK PRE00550

DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,4.E-03/ PRE00560

DATA OWFC/1.,.95,.5,.2,1.E-03/ PRE00570

TINC=1.0 PRE00580

TPRES=0.0 PRE00590

ICOUNT=0 PRE00600

DEL=.001 PRE00610

T(3)=0.0 PRE00620

SUM2=0.0 PRE00630

FIND AVERAGE OF THE DISC RELEASE HEIGHTS AND THE AVERAGE OF THE
INITIAL SPREADS OF THE DISCS PRE00640

DO 5 J=1,NDSCS PRE00650

SUM2=SUM2+R2DSC(J) PRE00660

5 CONTINUE PRE00670

```

ZREF=2.0
R0H2=SUM2/NDSCS
10 ICOUNT=ICOUNT+1
DO 50 I=1,3
T(I)=T(3)+FLOAT(I-1)*TINC

C FIND THE AVERAGE OF THE MOMENTS AT HEIGHT ZREF
      CALL AVRG(ZREF,T(I),QTOT,XBAVRG,SIG2X,SIG2Y)
      IF(QTOT.LT.1.E-10)GO TO 15
      SIGX=SQRT(SIG2X+R0H2/2.)
      SIGY=SQRT(SIG2Y+R0H2/2.)
      FRB(I)=SQRT(SIGX*SIGY)*1.5
      XBC(I)=XBAVRG
      GO TO 20
15 CALL WIN(ZREF,UX,V)
      XBC(1)=UX*T(I)
      FRB(I)=0.0
20 IF(T(I).GT.TWIND)GO TO 30
      CALL RISE(TPRES,T(I),DEL)
      F22(I)=ZCM+(2./3.)*RSPH
      IF(F22(I).GT.ZINV)F22(I)=ZINV
      FRT(I)=RSPH
      XBC(2)=XCM
      GO TO 40
30 XBC(2)=XTR+VTR*(T(I)-TTR)
      SIGXZ=SIG02+2.*KX*(T(I)-TTR)
      SIGZZ=SIG02+2.*KZ*(T(I)-TTR)
      SIGX=SQRT(SIGXZ)
      SIGZ=SQRT(SIGZZ)
      SIG=SQRT(SIGX*SIGZ)
      F22(I)=ZTR+SIG
      FRT(I)=1.5*SIG
40 FXC1(I)=(XBC(2)-XBC(1))/(F22(I)-ZREF)
      FXC0(I)=XBC(1)-FXC1(I)*ZREF
50 CONTINUE

C COMPUT AND STORE QUADRATIC FITS
      TIMES(ICOUNT)=T(3)

C FIT AND STORE RADIUS AT TOP
      CALL FIT(T,FRT,A,B,C)
      RT(1,ICOUNT)=A
      RT(2,ICOUNT)=B
      RT(3,ICOUNT)=C

C FIT AND STORE RADIUS AT BOTTOM
      CALL FIT(T,FRB,A,B,C)
      RB(1,ICOUNT)=A
      RB(2,ICOUNT)=B
      RB(3,ICOUNT)=C

C FIT AND STORE HEIGHT OF CLOUD
      CALL FIT(T,F22,A,B,C)
      Z2(1,ICOUNT)=A
      Z2(2,ICOUNT)=B
      Z2(3,ICOUNT)=C

C FIT AND STORE XC1
      CALL FIT(T,FXC1,A,B,C)
      XC1(1,ICOUNT)=A
      XC1(2,ICOUNT)=B
      XC1(3,ICOUNT)=C

C FIT AND STORE XC0

```

```

C          CALL FIT(T,FXC0,A,B,C)          PRE01410
C          XC0(1,ICOUNT)=A              PRE01420
C          XC0(2,ICOUNT)=B              PRE01430
C          XC0(3,ICOUNT)=C              PRE01440
C          PRE01450
C          PRE01460
C          PRE01470
C          PRE01480
C          PRE01490
C          PRE01500
C          PRE01510
C          NWL=1                      PRE01520
C          ACL=(RCARB1*OWFC(NWL,NSOIL)+RCARB2*OWFC(NWL))*VLOAD*PLEN/VOL  PRE01530
C          TINC=1.2*TINC               PRE01540
C          IF(ICOUNT.LT.25.AND.AC>..001)GO TO 10  PRE01550
C          PRE01560
C          PRE01570
C          PRE01580
C          PRE01590
C          WRITE(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)  PRE01600
C          WRITE(IFILE)DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV  PRE01610
C          WRITE(IFILE)VLOAD,RCARB1,RCARB2               PRE01620
C          WRITE(IFILE)ICOUNT               PRE01630
C          DO 1000 J=1,ICOUNT             PRE01640
C          WRITE(IFILE) TIMES(J),(RT(I,J),RB(I,J),Z2(I,J),XC0(I,J),  PRE01650
C          ,           XC1(I,J),I=1,3)               PRE01660
1000 CONTINUE          PRE01670
REWIND IFILE          PRE01680
999 RETURN            PRE01690
END                  PRE01700

```

```

SUBROUTINE PRETRN(TRN,REC,TIME,TRNLOS) PRT00220
      COMPUTE THE TRANSMITTANCE FOR THE RANDOM IN SPACE AND TIME PRT00010
      DISTRIBUTION OF CHARGES PRT00020
      INPUTS PRT00030
      TRN -TRANSMITTER COORDINATES IN LOCAL COORDINATE SYSTEM PRT00040
      REC -RECEIVER COORDINATES IN LOCAL COORDINATE SYSTEM PRT00050
      TIME -TIME AT WHICH TRANSMITTANCE IS DESIRED PRT00060
      NWL -INTEGER INDEX FOR WAVELENGTH PRT00070
      NSOIL -SOIL TYPE PRT00080
      OUTPUTS PRT00090
      TRNLOS -TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT PRT00100
*****
LOGICAL TEST PRT00110
***** PRT00120
***** PRT00130
***** PRT00140
***** PRT00150
***** PRT00160
***** PRT00170
***** PRT00180
***** PRT00190
***** PRT00200
***** PRT00210
***** PRT00220
***** PRT00230
***** PRT00240
***** PRT00250
***** PRT00260
***** PRT00270
***** PRT00280
***** PRT00290
***** PRT00300
***** PRT00310
***** PRT00320
***** PRT00330
***** PRT00340
***** PRT00350
***** PRT00360
***** PRT00370
***** PRT00380
***** PRT00390
***** PRT00400
***** PRT00410
***** PRT00420
***** PRT00430
***** PRT00440
***** PRT00450
***** PRT00460
***** PRT00470
***** PRT00480
***** PRT00490
***** PRT00500
***** PRT00510
***** PRT00520
***** PRT00530
***** PRT00540
***** PRT00550
***** PRT00560
***** PRT00570
***** PRT00580
***** PRT00590
***** PRT00600
***** PRT00610
***** PRT00620
***** PRT00630
***** PRT00640
***** PRT00650
***** PRT00660
***** PRT00670
***** PRT00680
***** PRT00690
***** PRT00700
C     PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
C
TEST=.FALSE.
XNORM=0.0
DO 10 I=1,3
RE(I)=REC(I)
TR(I)=TRN(I)
U(I)=RE(I)-TR(I)
XNORM=XNORM+U(I)**2
10 CONTINUE
XNORM=SQRT(XNORM)
U(1)=U(1)/XNORM
U(2)=U(2)/XNORM
U(3)=U(3)/XNORM
C     COMPUTE THE CONTRIBUTION FROM EACH CHARGE TO THE OPTICALLY
C     WEIGHTED CONCENTRATION ALONG THE LINE OF SIGHT
C
SUM=0.0
DO 100 I=1,ITOT
IF(TIME.LT.TSTAG(I))GO TO 100
TOF=TIME-TSTAG(I)
IF(TOF.GT.TIMES(ICOUNT))GO TO 100
DO 20 J=1,ICOUNT
IND=J
IF(TOF.LE.TIMES(J))GO TO 30
20 CONTINUE
C     DETERMINE NECESSARY PARAMETERS DESCRIBING THE CONICAL SHAPE SO THAT
C     THE LENGTH OF INTERSECTION OF THE LINE OF SIGHT AND CONE CAN BE
C     DETERMINED
C
30 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)
X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)
HTTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)

```

```

RTOP=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND) PRT00710
HTBOT=0.0 PRT00720
RBOT=TOF*(RB(1,IND)*TOF+RB(2,IND))+RB(3,IND) PRT00730
XCEN=DIFF(1,I)+(X1*HTTOP+X0) PRT00740
YCEN=DIFF(2,I) PRT00750
XB=DIFF(1,I)+(X1*HTBOT+X0) PRT00760
YB=DIFF(2,I) PRT00770
IF(ABS(U(3)),LT.1.E-06)GO TO 40 PRT00780
C COMPUTE THE INTERSECTION LENGTH FOR A NON-HORIZONTAL LINE OF SIGHT PRT00790
C CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB, PRT00800
1 XNORM,PLEN) PRT00810
GO TO 50 PRT00820
C COMPUTE THE INTERSECTION LENGTH FOR A HORIZONTAL LINE OF SIGHT PRT00830
PRT00840
C 40 IF(HTTOP,LT,TR(3))GO TO 45 PRT00850
A=U(1)**2+U(2)**2 PRT00860
B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN) PRT00870
C DETERMINE THE RADIUS,X,AND Y POSITIONS OF THE CONE AT THE PRT00880
TRANSMITTER HEIGHT PRT00890
PRT00900
ZETA=TR(3)/HTTOP PRT00910
RAD=ZETA*RTOP+(1.-ZETA)*RBOT PRT00920
XCEN=ZETA*XCEN+(1.-ZETA)*XB PRT00930
YCEN=ZETA*YCEN+(1.-ZETA)*YB PRT00940
PRT00950
A=U(1)**2+U(2)**2 PRT00960
B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN) PRT00970
C=(TR(1)-XCEN)**2+(TR(2)-YCEN)**2-RAD**2 PRT00980
PRT00990
X=B**2-A*C PRT01000
IF(X,LT.0)GO TO 45 PRT01010
P1=(-B+SQRT(X))/A PRT01020
P2=(-B-SQRT(X))/A PRT01030
PRT01040
IF(P1.GT.XNORM.AND.P2.GT.XNORM)GO TO 45 PRT01050
IF(P2.LT.0.0.AND.P1.LT.0.0)GO TO 45 PRT01060
PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0) PRT01070
PRT01080
GO TO 50 PRT01090
45 PLEN=0.0 PRT01100
50 VOL=(PI/3.)*(HTTOP-HTBOT)*(RTOP**2+RTOP*RBOT+RBOT**2) PRT01110
CONT=VLOAD*PLEN/VOL PRT01120
SUM=SUM+CONT PRT01130
ACLSKT=0.0 PRT01140
ACLSPH=0.0 PRT01150
CALL TRNCHK(ACLSKT,SUM,ACLSPH) PRT01160
IF(TEST)GO TO 998 PRT01170
100 CONTINUE PRT01180
TRNLDS=EXP(-SUM*(RCARB1*OWFC(NWL,NSOIL)+RCARB2*OWFC(NWL))) PRT01190
GO TO 999 PRT01200
998 TRNLDS=0.0 PRT01210
999 RETURN PRT01220
END PRT01230

```

```

SUBROUTINE PREVEH(NSOIL,NWL) PRV00360
C ROUTINE FOR PRECOMPUTING VEHICLE GENERATED DUST CLOUD PRV00010
C
C INPUTS PRV00020
C NSOIL - SOIL TYPE (SEE DRTRAN FOR DETAILS) PRV00030
C NWL - WAVELENGTH INDEX (SEE DRTRAN) PRV00040
C
C OUTPUTS PRV00050
C RT - COEFFICIENTS OF QUADRATIC FIT TO RADIUS OF THE CLOUD PRV00060
C Z2 - COEFFICIENTS OF QUADRATIC FIT TO HEIGHT OF THE CLOUD PRV00070
C XC1 - COEFFICIENTS OF QUADRATIC FIT TO SLOPE OF LINE DESCRIBING PRV00080
C X DISPLACEMENT OF CONE (ALONGWIND) PRV00090
C XC0 - COEFFICIENTS OF QUADRATIC FIT TO CONSTANT TERM OF LINE PRV00100
C DESCRIBING X DISPLACEMENT OF THE CONE (ALONGWIND) PRV00110
C
C THE ABOVE OUTPUT ARE THE COEFFICIENTS OF QUADRATIC FITS THROUGH PRV00120
C THREE CONSECUTIVE POINTS IN TIME. THE QUADRATIC FITS ARE STORED IN PRV00130
C COMMON/ M05 / WITH THE ARRAY TIMES CONTAINING THE LAST TIME OF THAT PRV00140
C INTERVAL. THE FITS ARE WRITTEN ONTO A FILE INDICATED BY IFILE USING PRV00150
C A BINARY WRITE. PRV00160
C
C THE FITS ARE STORED SUCH THAT PRV00170
C
C F<TIME>=VAR<1,J>*TIME**2 + VAR<2,J>*TIME + VAR<3,J> PRV00180
C AND TIMES<J-1> < TIME < TIMES<J> PRV00190
C
C ****LOGICAL HORIZ**** PRV00200
C DIMENSION OWF(5,2),T(3),FR(3),FZ2(3) PRV00210
C DIMENSION FXC1(3),FXC0(3),XB(2) PRV00220
C COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3) PRV00230
C COMMON/PRE/ZT0,RT2DZ PRV00240
C COMMON/MODE/HORIZ PRV00250
C COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,DMM(600), PRV00260
C + ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25), PRV00270
C + RB(3,25),Z2(3,25) PRV00280
C COMMON /I0UNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIR TU,NCLIMT,KSTOR,NPLOTUP PRV00290
C COMMON/PRTINF/R0,VGRAV(3),NPRTS PRV00300
C COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2) PRV00310
C COMMON/CONST/PI,P12,PIRAD,TWOP1,TORRMB,CDEGK PRV00320
C
C DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/ PRV00330
C
C SET UP CORRECT PARAMETERS SO THAT CWIND CAN BE USED PRV00340
C
C NARY=1 PRV00350
C ITOT=1 PRV00360
C NTOT=1 PRV00370
C NCHTOT=1 PRV00380
C HORIZ=.TRUE. PRV00390
C ONEM=-1. PRV00400
C COSTH=1.0 PRV00410
C SINTH=0.0 PRV00420
C COSTH2=COSTH**2 PRV00430
C SINTH2=SINTH**2 PRV00440
C SCRN(1)=SINTH PRV00450
C SCRN(2)=-COSTH PRV00460
C X=0.0 PRV00470
C Y=0.0 PRV00480
C TMAX=1000. PRV00490
C
C

```

```

C COMPUTE R, Z2, XC1, XCO AT THREE CONSECUTIVE TIMES WITH SPACING      PRV00710
C TINC AND THEN CALL FIT, WHICH CALCULATES A QUADRATIC FIT TO THESE      PRV00720
C POINTS AND STORE THEM IN COMMON /QUADFT/                                PRV00730
C
C      TINC=1.0
C      ICOUNT=0
C      T(3)=0.0
C 10 ICOUNT=ICOUNT+1
C      DO 20 I=1,3
C      T(I)=T(3)+FLOAT(I-1)*TINC
C      ZREF=2T0+SQRT(T(I))*RT2DZ
C      TOF=T(I)-TDSC(NDSCE)
C      CALL MOMENT(VGRAV,ZREF,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)          PRV00740
C
C COMPUTE R THE RADIUS OF THE CLOUD                                     PRV00750
C
C      ROH2=R2DSC(1)
C      SIGW=SQRT(SIGW2+ROH2/2.)
C      SIGP=SQRT(SIGP2+ROH2/2.)
C      FR(I)=1.5*SQRT(SIGW*SIGP)
C      ACL=CWIND(X,Y,ZREF,T(I))*OWF(NWL,NSOIL)                         PRV00760
C
C COMPUTE Z2 APPROXIMATE HEIGHT OF THE CLOUD                           PRV00770
C
C      FZ2(I)=(2.*QDSC(1,1))/PI/FR(I)/ACL                            PRV00780
C      Z22=FZ2(I)                                                       PRV00790
C
C COMPUTE THE X POSITION OF THE CLOUD AT A HEIGHT OF Z2 AND A HEIGHT      PRV00800
C OF 1 METER.                                                               PRV00810
C
C      CALL MOMENT(VGRAV,Z22,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)           PRV00820
C      XB(2)=XBAR+XDSC(1)
C      Z1=1.0
C      CALL MOMENT(VGRAV,Z1,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)           PRV00830
C      XB(1)=XBAR+XDSC(1)
C      FXC1(I)=(XB(2)-XB(1))/(Z22-Z1)                                 PRV00840
C      FXC0(I)=XB(1)-FXC1(I)                                         PRV00850
C 20 CONTINUE
C
C COMPUTE AND STORE THE QUADRATIC FITS                                  PRV00860
C
C      TIMES(ICOUNT)=T(3)
C
C FIT AND STORE THE CLOUD RADIUS                                       PRV00870
C
C      CALL FIT(T,FR,A,B,C)
C      RT(1,ICOUNT)=A
C      RT(2,ICOUNT)=B
C      RT(3,ICOUNT)=C
C
C FIT AND STORE Z2, APPROXIMATE CLOUD HEIGHT                          PRV00880
C
C      CALL FIT(T,FZ2,A,B,C)
C      Z2(1,ICOUNT)=A
C      Z2(2,ICOUNT)=B
C      Z2(3,ICOUNT)=C
C
C FIT AND STORE XC1
C
C      CALL FIT(T,FXC1,A,B,C)
C      XC1(1,ICOUNT)=A
C      XC1(2,ICOUNT)=B
C      XC1(3,ICOUNT)=C
C
C FIT AND STORE XCO
C
C      CALL FIT(T,FXC0,A,B,C)
C      XC0(1,ICOUNT)=A
C      XC0(2,ICOUNT)=B
C      XC0(3,ICOUNT)=C

```

```
TINC=i,2*TINC  
IF(ICOUNT.LT.20.AND.T(3).LT.TMAX) GO TO 10  
RETURN  
END
```

```
PRV01410  
PRV01420  
PRV01430  
PRV01440
```

```

SUBROUTINE RISE(TPRES,TNEXT,DEL) RIS00010
REAL M,NDIF,KZ,KX RIS00020
LOGICAL SWITCH,CHANGE RIS00030
DIMENSION WK(12,6) RIS00040
COMMON /BUOYCL/ Y(8),SPHNS(3),RISTIM RIS00050
COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,M,NDIF,ZINV RIS00060
COMMON /CLOCK/ TIME,TWIND RIS00070
COMMON /STARS/ USTAR,TSTAR,ZSTAR RIS00080
COMMON /EKTEMP/ ZL,T0,TC1,TC2,TC3 RIS00090
COMMON /TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE RIS00100
COMMON /SIG/SIG02,SIGC RIS00110
COMMON /IOUNIT/IOUT,IOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUR RIS00120
COMMON /DISCS/ NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20), RIS00130
10DSC(20,3) RIS00140
COMMON /BURST/ ACCEL,TBURST RIS00150
DATA HMIN,ACCURC,WK,N,ND/.001,.001,72*0,,8,12/ RIS00160
C ****

```

PURPOSE

THIS ROUTINE CALLS A RUNGA KUTTA ROUTINE TO INTEGRATE IN TIME THE EQUATIONS FOR THE RISE OF A BUOYANT CLOUD BEGINNING AT TPRES ENDING AT TNEXT UNLESS THE CONDITION FOR SWITCHING TO THE WIND DISPERSION MODEL IS ENCOUNTERED IN WHICH CONVRT IS CALLED. SEE SUBROUTINE DIFEQ FOR THE DEFINITIONS OF Y(I).

ARGUMENTS

TPRES	AS INPUT TPRES IS THE INITIAL TIME OF THIS SEGMENT OF INTEGRATION AND IS RETURNED WITH THE VALUE OF THE LAST SUCCESSFUL INTEGRATION STEP.	RIS00310
TNEXT	THE ENDPOINT OF THE TIME INTERVAL WHICH IS INPUT.	RIS00330

REQUIRED SUBROUTINES

RKM	A RUNGA-KUTTA-MERSON INTEGRATION ROUTINE	RIS00380
CONVRT	A SUBROUTINE WHICH CONVERTS THE CURRENT BUOYANT DUST CLOUD TO A NUMBER OF DISC SOURCES FOR THE WIND DISPERSION MODEL. A GAP TIME DURING WHICH THE BUOYANT MODEL IS CONTINUED IS COMPUTED.	RIS00400
WNDCAL	COMPUTES SCALED WIND SPEED AT A SPECIFIED HEIGHT	RIS00410
DIFFUS	COMPUTES DIFFUSIVITY AT A SPECIFIED HEIGHT	RIS00420

CALLED BY DUSTCL

```

***** IF(TNEXT.GT.TWIND)GO TO 999
SWITCH=.FALSE.
CHANGE=.FALSE.
T2=TPRES

```

PERFORM INTEGRATION IN SEGMENTS OF TIME

```

10 DO 20 NT=1,300
T1=T2
T2=1.2*T1
IF(T2.LE.0.)T2=.5
IF(T2.GT.TNEXT)T2=TNEXT
IF (DEL.LT.HMIN)DEL=HMIN
CALL RKM(N,T1,T2,Y,HMIN,DEL,ACCURC,WK,ND)

```

CHECK TO SEE IF CLOUD GROWTH IS DOMINATED BY WIND DIFFUSION OVER BUOYANT RISE BY COMPARING WIND DIFFUSIVITY, DIFW, TO THE EFFECTIVE BUOYANT DIFFUSIVITY, DIFB AND IF THE HEIGHT OF THE CENTER OF MASS IS LESS THAN ZSTAR SWITCH TO THE

```

C      WIND MODEL.                                RIS00710
C
5   IF(T2.LT.TBURST)GO TO 15                   RIS00720
    DIFB=ABS(.1111*Y(1)*Y(3))                  RIS00730
    TOP=Y(6)+Y(1)                               RIS00740
    DIFW=DIFFUS(Z0,ZL, TOP)                     RIS00750
    IF(DIFB.GT.DIFW)GO TO 15                   RIS00760
    IF(TOP.GT.ZSTAR)SWITCH=.TRUE.               RIS00770
    IF(TOP.GT.ZSTAR)GO TO 15                   RIS00780
    VTR=USTAR*WNDCAL(Z0,ZL,Y(6))              RIS00790
    KZ=DIFFUS(Z0,ZL,Y(6))                      RIS00810
    KX=DXZ0*KZ                                  RIS00820
    TTR=T2                                     RIS00830
    XTR=Y(4)                                    RIS00840
    ZTR=Y(6)                                    RIS00850
    CHANGE=.TRUE.                               RIS00860
    CALL CONVRT(T2)                            RIS00870
    GO TO 200                                  RIS00880
15  CONTINUE                                 RIS00890
    IF(T2.GE.TNEXT)GO TO 200                  RIS00900
    IF(T2.GT.300,)GO TO 99                    RIS00910
20  CONTINUE                                 RIS00920
99  WRITE(I00UT,98)                          RIS00930
98  FORMAT(54H *** DIRTRAN ERROR - 5 MINUTE CUT-OFF ON BUOYANT RISE ) RIS00940
    STOP                                     RIS00950
200 TPRES=T2                                RIS00960
    RISTIM=TPRES                             RIS00970
999 RETURN                                  RIS00980
    END                                      RIS00990

```

```

SUBROUTINE RKM(N,XL,XU,Y,HMIN,DEL,ACCURC,WK,ND)          RKM00010
C                                                               RKM00020
C                                                               RKM00030
C                                                               RKM00040
C                                                               RKM00050
C                                                               RKM00060
C                                                               RKM00070
C                                                               RKM00080
C                                                               RKM00090
C                                                               RKM00100
C                                                               RKM00110
C                                                               RKM00120
C                                                               RKM00130
C                                                               RKM00140
C                                                               RKM00150
C                                                               RKM00160
C                                                               RKM00170
C                                                               RKM00180
C                                                               RKM00190
C                                                               RKM00200
C                                                               RKM00210
C                                                               RKM00220
C                                                               RKM00230
C                                                               RKM00240
C                                                               RKM00250
C                                                               RKM00260
C                                                               RKM00270
C                                                               RKM00280
C                                                               RKM00290
C                                                               RKM00300
C                                                               RKM00310
C                                                               RKM00320
C                                                               RKM00330
C                                                               RKM00340
C                                                               RKM00350
C                                                               RKM00360
C                                                               RKM00370
C                                                               RKM00380
C                                                               RKM00390
C                                                               RKM00400
C                                                               RKM00410
C                                                               RKM00420
C                                                               RKM00430
C                                                               RKM00440
C                                                               RKM00450
C                                                               RKM00460
C                                                               RKM00470
C                                                               RKM00480
C                                                               RKM00490
C                                                               RKM00500
C                                                               RKM00510
C                                                               RKM00520
C                                                               RKM00530
C                                                               RKM00540
C                                                               RKM00550
C                                                               RKM00560
C                                                               RKM00570
C                                                               RKM00580
C                                                               RKM00590
C                                                               RKM00600
C                                                               RKM00610
C                                                               RKM00620
C                                                               RKM00630
C                                                               RKM00640
C                                                               RKM00650
C                                                               RKM00660
C                                                               RKM00670
C                                                               RKM00680
C                                                               RKM00690
C                                                               RKM00700

NUMERICAL INTEGRATION ROUTINE FOR SYSTEMS OF ODE'S          USING THE RUNGE-KUTTA-MERSON TECHNIQUE

INPUT PARAMETERS

N - NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS          RKM00100
XL - INITIAL ABCISSA OF THE INTERVAL                      RKM00110
XU - THE FINAL ABCISSA OF THE INTEGRATION INTERVAL        RKM00120
Y - A SINGLY DIMENSIONED ARRAY OF LENGTH N. WHEN          RKM00130
     RKM IS CALLED IT MUST CONTAIN THE VALUES OF          RKM00140
     THE DEPENDENT VARIABLES AT XL. UPON RETURN          RKM00150
     TO THE CALLING PROGRAM Y CONTAINS THE VALUES          RKM00160
     OF THE DEPENDENT VARIABLES AT XU.                  RKM00170
HMIN - THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE    RKM00180
     INTEGRATION.                                         RKM00190
DEL - THE INITIAL ESTIMATE OF THE STEP SIZE AND UPON      RKM00200
     RETURN TO THE CALLING PROGRAM DEL CONTAINS THE      RKM00210
     FINAL STEP SIZE USED. THIS VALUE SHOULD BE USED      RKM00220
     IN THE NEXT CALL TO PRODUCE AN EFFICIENT INTEGRATION. RKM00230
     DEL IS RETURNED WITH THE VALUE ZERO IF IT HAS       RKM00240
     BEEN HALVED BELOW HMIN.                            RKM00250
ACCURC - PREASSIGNED ACCURACY WHICH IS ALSO USED IN ADJUSTING RKM00260
     THE STEP SIZE.                                         RKM00270
WK - AT LEAST A BLOCK OF N BY 6 FLOATING POINT LOCATIONS RKM00280
     USED FOR A WORK ARRAY.                                RKM00290
ND - THE DIMENSION OF ARRAYS Y AND WK.                   RKM00300

IT IS REQUIRED THAT THE USER OF RKM WRITE A SUBROUTINE      RKM00310
DEFINING THE DIFFERENTIAL EQUATIONS. THE SUBROUTINE        RKM00320
STATEMENT SHOULD LOOK LIKE - SUBROUTINE DIFEQ(N,X,Y,YP) . RKM00330
RKM00340
RKM00350
RKM00360
RKM00370
RKM00380
RKM00390
RKM00400
RKM00410
RKM00420
RKM00430
RKM00440
RKM00450
RKM00460
RKM00470
RKM00480
RKM00490
RKM00500
RKM00510
RKM00520
RKM00530
RKM00540
RKM00550
RKM00560
RKM00570
RKM00580
RKM00590
RKM00600
RKM00610
RKM00620
RKM00630
RKM00640
RKM00650
RKM00660
RKM00670
RKM00680
RKM00690
RKM00700

WHERE
N - THE NUMBER OF EQUATIONS
X - THE INDEPENDENT VARIABLE
Y - SINGLY DIMENSIONED ARRAY OF DEPENDENT VARIABLES
YP - SINGLY DIMENSIONED ARRAY OF THE RATES OF Y AT X
     YP(I) = D Y(I)/DX

DIMENSION Y(ND),WK(ND,6)
LOGICAL FIRST,QUIT

SET UP NEEDED VARIABLES UPON ENTRY

XN=XL
H=DEL
FIRST=.TRUE.
QUIT=.FALSE.

CHECK IF XN IS CLOSE TO XU
20 IF(XN+H .LT. XU) GO TO 30
DEL=H
H=XU-XN
QUIT=.TRUE.
IF(FIRST) DEL=H

MAKE FIRST CALL TO DIFEQ AT THE BEGINNING OF INTERVAL
30 CALL DIFEQ(N,XN,Y,WK(1,1))

PERFORM THE RUNGE-KUTTA-MERSON ALGORITHM
40 H3=H/3,
DO 50 I=1,N

```

```

50 WK(I,3)=H3*WK(I,1) RKM00710
50 WK(I,6)=Y(I)+WK(I,3) RKM00720
CALL DIFEQ(N,XN+H3,WK(1,6),WK(1,2)) RKM00730
DO 60 I=1,N RKM00740
60 WK(I,6)=Y(I)+(WK(I,3)+H3*WK(I,2))/2. RKM00750
CALL DIFEQ(N,XN+H3,WK(1,6),WK(1,2)) RKM00760
DO 70 I=1,N RKM00770
WK(I,4)=H3*WK(I,2) RKM00780
70 WK(I,6)=Y(I)+(3.*WK(I,3)+9.*WK(I,4))/8. RKM00790
CALL DIFEQ(N,XN+H/2.,WK(1,6),WK(1,2)) RKM00800
DO 80 I=1,N RKM00810
WK(I,5)=H3*WK(I,2) RKM00820
80 WK(I,6)=Y(I)+(3.*WK(I,3)-9.*WK(I,4)+12.*WK(I,5))/2. RKM00830
CALL DIFEQ(N,XN+H,WK(1,6),WK(1,2)) RKM00840
RKM00850
C FIND THE LARGEST RELATIVE ERROR RKM00860
C TEST=0, RKM00870
DO 90 I=1,N RKM00880
YX=Y(I) RKM00890
IF(YX .EQ. 0.) YX=ACURC RKM00900
E=((WK(I,3)-9.*WK(I,4)/2.+4.*WK(I,5)-H3*WK(I,2)/2.)/5.)/YX RKM00910
90 TEST=AMAX1(TEST,ABS(E)) RKM00920
FIRST=.FALSE. RKM00930
IF(TEST .LT. ACCURC) GO TO 100 RKM00940
RKM00950
C IF THE LARGEST ERROR IS GREATER THAN ACCURC HALF THE STEP RKM00960
C SIZE AND TRY AGAIN. RKM00970
C H=H/2. RKM00980
IF(H .LT. HMIN) GO TO 10 RKM01000
QUIT=.FALSE. RKM01010
GO TO 40 RKM01020
RKM01030
C TRUNCATION ERROR LESS THAN ACCURC, RESET THE Y ARRAY TO RKM01040
SET UP FOR THE NEXT INTERVAL RKM01050
C 100 XN=XN+H RKM01060
DO 110 I=1,N RKM01070
110 Y(I)=Y(I)+(WK(I,3)+4.*WK(I,5)+H3*WK(I,2))/2. RKM01080
RKM01090
C CHECK FOR STEP SIZE DOUBLING. DOUBLE IF LARGEST RELATIVE RKM01100
C ERROR IS 32 TIMES LESS THAN ACCURC. RKM01120
C IF(.NOT.(TEST .GE. ACCURC/32. .OR. QUIT)) H=H+H RKM01130
IF(.NOT. QUIT) GO TO 20 RKM01140
RETURN RKM01150
RKM01160
C THE VALUE OF H (DEL) IS LESS THAN THE SPECIFIED MINIMUM. RKM01170
C REPORT THIS AND ERROR OUT. RKM01180
RKM01190
C 10 CONTINUE RKM01200
1000 FORMAT('1 H BELOW HMIN''/0      INTEGRATION ABORTED') RKM01210
DEL=0, RKM01220
RETURN RKM01230
END RKM01240
RKM01250
RKM01260

```

```

SUBROUTINE SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)           SET00010
  DIMENSION NCHS(2),SRCBAS(2),SIDE1(2),SIDE2(2),TRNFRM(2,2),REF(2) SET00020
  COMMON /ARRAY/DYRLAP,AREA,PERIM,PRJARY,CENDIF              SET00030
  COMMON/M05/DIFF(2,200),NCHTOT,PRSEPC(200),NTOT,NARY,ITOT,      SET00040
  + COOR(2,200),TSTAG(200),DMMY(401)                      SET00050
  COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSSET00060
C*****SET00070
CCCCC PURPOSE SET00080
TO CONVERT THE USER DEFINED COORDINATES OF THE CHARGES TO THE SET00090
LOCAL COORDINATE SYSTEM. SET00100
CCCCC INPUTS SET00110
NCHS - SINGLY DIMENSIONED ARRAY CONTAINING THE NUMBER OF SET00120
CHARGES. SET00130
SRCBAS - A REFERENCE CHARGE IN THE USER DEFINED COORDINATES. SET00140
SET00150
SIDE1,SIDE2 - VECTORS DESCRIBING THE BOUNDING PARRALELOGRAM. SET00160
SET00170
TRNFRM - COORDINATE SYSTEM TRANSFORMATION MATRIX. SET00180
SET00190
CCCCC OUTPUTS RETURNED IN COMMON /ARRAY/ AND /SEPRTN/ SET00200
SET00210
DIFF- DOUBLY DIMENSIONED ARRAY CONTAINING THE CHARGE COORDINATESSET00220
IN THE LOCAL COORDINATE SYSTEM. SET00230
SET00240
ITOT- TOTAL NUMBER OF CHARGES. SET00250
SET00260
NCHTOT- WHEN NARY=1 OR 2 THE TOTAL NUMBER OF CHARGES. WHEN SET00270
WHEN NARY=3 IS SET =1. SET00280
SET00290
NTOT- WHEN NARY=1 OR 2 IS SET =1 AND WHEN NARY=3 IS THE TOTAL SET00300
NUMBER OF CHARGES. SET00310
SET00320
SET00330
SET00340
SET00350
SET00360
SET00370
SET00380
SET00390
SET00400
CCCCC SUBROUTINES AND FUNCTIONS SET00410
UNIT COMPUTES THE UNIT VECTOR OF A GIVEN VECTOR SET00420
C*****SET00430
IF(NARY.NE.1)GO TO 4 SET00440
NCHTOT=NCHS(1)*NCHS(2) SET00450
NTOT=1 SET00460
ITOT=NCHTOT SET00470
GO TO 6 SET00480
4 IF(NARY.NE.2)GO TO 5 SET00490
NCHTOT=NCHS(1) SET00500
NTOT=1 SET00510
ITOT=NCHTOT SET00520
GO TO 6 SET00530
5 IF(NARY.NE.3)GO TO 998 SET00540
NCHTOT=1 SET00550
NTOT=NCHS(1) SET00560
ITOT=NTOT SET00570
6 CONTINUE SET00580
SET00590
CCCCC DETERMINE THE COORDINATE OF THE REFERENCE CHARGE IN THE INTERNAL SET00600
COORDINATE SYSTEM SET00610
DO 20 I=1,2 SET00620
REF(I)=0.0 SET00630
DO 10 J=1,2 SET00640
REF(I)=REF(I)+TRNFRM(I,J)*SRCBAS(J) SET00650
10 CONTINUE SET00660
20 CONTINUE SET00670
DO 40 I=1,2 SET00680
DO 30 J=1,ITOT SET00690
SET00700
)

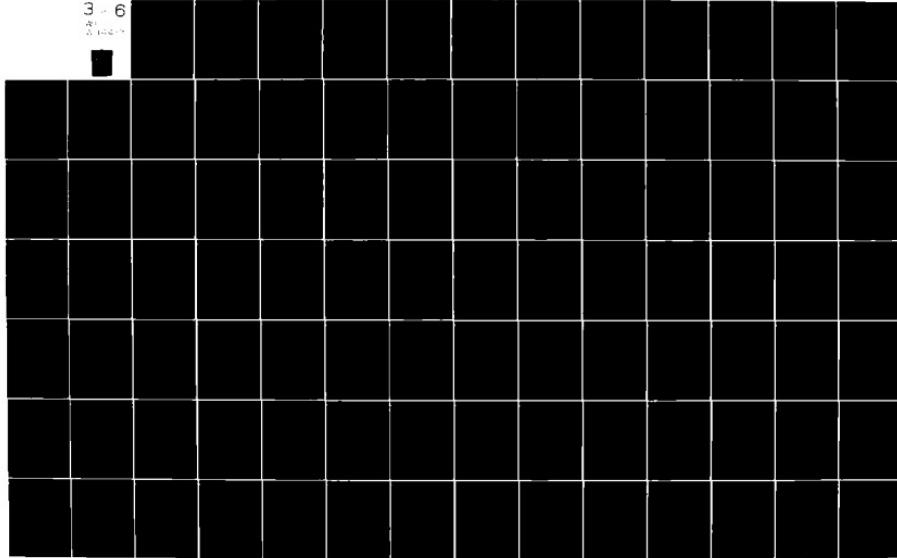
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G #1
PROGRAM LISTINGS FOR E0SAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(II)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU

NL

3 x 6

R G STEINHOFF



```

      DIFF(I,J)=0.0          SET00710
 30  CONTINUE          SET00720
 40  CONTINUE          SET00730
 IF(NARY.GT.1)GO TO 90    SET00740
 NM=0                  SET00750
 NC1=NCHS(1)            SET00760
 NC2=NCHS(2)            SET00770
 SET00780
C   COMPUTE LOCATIONS OF CHARGES FOR INTERNAL COORDINATE SYSTEM FOR
C   UNIFORMLY DISTRIBUTED CHARGES.          SET00790
SET00800
C
      DO 80 M=1,NC2        SET00810
      DO 70 N=1,NC1        SET00820
      NM=NM+1              SET00830
      DO 60 I=1,2          SET00840
      DO 50 J=1,2          SET00850
      DIFF(I,NM)=DIFF(I,NM)+FLODAT(N-1)*TRNFRM(I,J)*SIDE1(J)
      1 +FLODAT(M-1)*TRNFRM(I,J)*SIDE2(J)          SET00860
      SET00870
      SET00880
 50  CONTINUE          SET00890
 60  CONTINUE          SET00900
 70  CONTINUE          SET00910
 80  CONTINUE          SET00920
 GO TO 999              SET00930
 90  CONTINUE          SET00940
 SET00950
C   TRANSFORM CHARGE LOCATIONS TO LOCAL COORDINATE SYSTEM FOR RANDOM
C   CHARGES.          SET00960
SET00970
C
      NC1=NCHS(1)          SET00980
      DO 120 M=1,NC1        SET00990
      DO 110 I=1,2          SET01000
      DO 100 J=1,2          SET01010
      DIFF(I,M)=DIFF(I,M)+TRNFRM(I,J)*COOR(I,J,M)          SET01020
      SET01030
 100 CONTINUE          SET01040
 100 DIFF(I,M)=DIFF(I,M)-REF(I)          SET01050
 110 CONTINUE          SET01060
 120 CONTINUE          SET01070
 GO TO 999              SET01080
 998 WRITE(100UT,778)          SET01090
 778 FORMAT(5X,23H *** NARY OUT OF RANGE  >          SET01100
 999 RETURN          SET01110
 END          SET01120

```

SUBROUTINE SOURCE(W,NCHRG,DD,NSOIL,DSOD) ***** SORC0010
 C ***** SORC0020
 ***** SORC0030
 ***** SORC0040
 ***** SORC0050
 ***** SORC0060
 ***** SORC0070
 ***** SORC0080
 ***** SORC0090
 ***** SORC0100
 ***** SORC0110
 ***** SORC0120
 ***** SORC0130
 ***** SORC0140
 ***** SORC0150
 ***** SORC0160
 ***** SORC0170
 ***** SORC0180
 ***** SORC0190
 ***** SORC0200
 ***** SORC0210
 ***** SORC0220
 ***** SORC0230
 ***** SORC0240
 ***** SORC0250
 ***** SORC0260
 ***** SORC0270
 ***** SORC0280
 ***** SORC0290
 ***** SORC0300
 ***** SORC0310
 ***** SORC0320
 ***** SORC0330
 ***** SORC0340
 ***** SORC0350
 ***** SORC0360
 ***** SORC0370
 ***** SORC0380
 ***** SORC0390
 ***** SORC0400
 ***** SORC0410
 ***** SORC0420
 ***** SORC0430
 ***** SORC0440
 ***** SORC0450
 ***** SORC0460
 ***** SORC0470
 ***** SORC0480
 ***** SORC0490
 ***** SORC0500
 ***** SORC0510
 ***** SORC0520
 ***** SORC0530
 ***** SORC0540
 ***** SORC0550
 ***** SORC0560
 ***** SORC0570
 ***** SORC0580
 ***** SORC0590
 ***** SORC0600
 ***** SORC0610
 ***** SORC0620
 ***** SORC0630
 ***** SORC0640
 ***** SORC0650
 ***** SORC0660
 ***** SORC0670
 ***** SORC0680
 ***** SORC0690
 ***** SORC0700

PURPOSE
 TO CALCULATE EXPLOSIVE DUST SOURCE TERM FOR THE
 DIRTRAN CODE

INPUT

W	THE WEIGHT OF THE CHARGE IN KG-TNT	SORC0100
DD	DETONATION DEPTH IN METERS	SORC0110
NSOIL	INTEGER SOIL INDEX	SORC0120
DSOD	DEPTH OF SOD IN METERS	SORC0130

OUTPUT (RETURNED IN COMMON /PRTINF/, /BUOYCL/ AND /CARB/)

R0	INITIAL CLOUD RADIUS IN METERS	SORC0140
VGRAV	SINGLY DIMENSIONED ARRAY CONTAINING OPTICALLY WEIGHTED AVERAGE SETTLING VELOCITIES FOR EACH SIZE RANGE IN THE PARTICLE DISTRIBUTION (METERS/SEC)	SORC0150
NPRTS	THE NUMBER OF SIZE RANGES IN THE PARTITIONING OF THE PARTICLE SIZE SPECTRUM	SORC0160
RSPH	THE INITIAL RADIUS OF THE CLOUD IN METERS	SORC0170
DELT	THE INITIAL DIFFERENCE IN TEMPERATURE BETWEEN THE CLOUD AND SURROUNDINGS (DEGREES KELVIN)	SORC0180
VZSPH	THE INITIAL VERTICAL VELOCITY OF THE CLOUD (M/S)	SORC0190
XCMSPH	INITIAL HORIZONTAL POSITION OF THE CLOUD (METERS)	SORC0200
YCMSPH	INITIAL Y POSITION OF THE CLOUD (METERS)	SORC0210
ZCMSPH	INITIAL HEIGHT OF THE CLOUD (METERS)	SORC0220
XTOP	INITIAL X POSITION OF THE TOP OF THE CLOUD (METERS)	SORC0230
YTOP	INITIAL Y POSITION OF THE TOP OF THE CLOUD (METERS)	SORC0240
RISTIM	TIME LAPSED SINCE DETONATION IN SECONDS	SORC0250
RCARB1	PORTION OF BUOYANT CLOUD WHICH IS DIRT PARTICLES	SORC0260
RCARB2	PORTION OF BUOYANT CLOUD WHICH IS CARBON PARTICLES	SORC0270

CALLED BY DUSTCL

SUBROUTINES AND FUNCTIONS

NONE

**** LOGICAL HORIZ_ONCE SORC0480
 DIMENSION CR(5,7),CD(5,7),OWML(3,4),OWSV(3,4),PRTTH(4) SORC0490
 DIMENSION S(3),BURHTR(5),WTRAT(5) SORC0500
 COMMON/PRTINF/ R0,VGRAV(3),NPRTS SORC0510
 COMMON/IOUNIT/IOUT,IOPEN,IPHFUN,LOUNIT,NDIR TU,NCLIMT,KSTOR,NPLOTU SORC0520
 COMMON/BUOYCL/ RSPH,DELT,VZSPH,XCMSPH,YCMSPH,ZCMSPH,XTOP,YTOP, * SORC0530
 * SPHNS(3),RISTIM SORC0540
 COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3 SORC0550
 COMMON/STARS/USTAR,TSTAR,ZSTAR SORC0560
 COMMON/WNDPRM/ DXZ0,DYX0,DZ0,U0,UM,DN,ZINV SORC0570
 COMMON/BURST/ ACCEL,TBURST SORC0580
 COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2) SORC0590
 COMMON/MODE/ HORIZ SORC0600
 COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
 1 QDSC(20,3) SORC0610
 COMMON/CARB/RCARB1,RCARB2 SORC0620
 COMMON/NTAL/TNOT,YOLSPH,TNO,CBLEED SORC0630
 CR IS THE CRATER RADIUS INDEXED BY COEFFICIENT AND SOIL TYPE SORC0640
 DATA CR/.271,-.684,.39,.886,0.,.271,-.684,.39,.886,0.,
 1 .386,-.849,.367,.993,0.,.503,-.954,.45,1.19,0.,
 2 .629,-1.08,.264,1.12,0.,.629,-1.08,.264,1.12,0., SORC0650
 SORC0660
 SORC0670
 SORC0680
 SORC0690
 SORC0700

```

3 .806,-1.28,-.178,.852,0./ SORC0710
C CD IS THE CRATER DEPTH INDEXED BY COEFFICIENT AND SOIL TYPE SORC0720
C DATA CD/.113,-.477,.27,1.84,1.05,.134,-.571,.343,2.24,1.31, SORC0730
C 1 .189,-.84,.447,3.3,2.1,.251,-1.17,.494,4.72,3.34, SORC0740
C 2 .189,-.84,.447,3.3,2.1,.331,-1.49,.579,4.92,3.13, SORC0750
C 3 .449,-1.82,.322,4.11,2.02/ SORC0760
C OWM IS THE OPTICALLY WEIGHTED MASS LOADING COEFFICIENT INDEXED BY SORC0770
C BIN SIZE AND SOIL TYPE SORC0780
C DATA OWM/.2.88E3,2*0.,3.08E3,8*0./ SORC0790
C OWSV IS THE OPTICALLY WEIGHTED PARTICLE SETTLING VELOCITY (CM/SEC) SORC0800
C INDEXED BY BIN SIZE AND SOIL TYPE SORC0810
C DATA OWSV/12*0./ SORC0820
C PRTTN IS THE PARTITIONING RATIO INDEXED ON SOIL TYPE SORC0830
C DATA PRTTN/4*.8/ SORC0840
C BURHTR IS THE RATIO OF BURST HEIGHT TO INITIAL RADIUS AND WTRAT SORC0850
C IS THE FRACTION OF THE TOTAL WEIGHT WHICH IS EFFECTIVE IN THE CLOUD SORC0860
C DATA BURHTR/0.,4.,2.,4.,3./,WTRAT/.6,1.,.8,1.,.7/ SORC0870
C
C RISTM=0, SORC0880
C XCMSPH=0, SORC0890
C YCMSPH=0, SORC0900
C XT0P=0, SORC0910
C YT0P=0, SORC0920
C TNO=T0 SORC0930
C NPRS=1 SORC0940
C SCARB IS THE OPTICALLY WEIGHTED CARBON PARTICLE LOADING COEFFICIENT SORC0950
C SCARB=270.*W SORC0960
C W3=(W*WTRAT(NCHRG))**.333333 SORC0970
C R0=2.0*w3 SORC0980
C TAMB=T0+TMPCAL(Z0,ZL,R0)*TSTAR SORC0990
C DELT=.57*TAMB SORC1000
C RSPH=R0 SORC1010
C ZCMSPH=R0 SORC1020
C BURHT=BURHTR(NCHRG)*R0 SORC1030
C BURVZ=1.3*SQRT(R0) SORC1040
C TBURST=.15*R0 SORC1050
C VZSPH=.2.*BURHT/TBURST-BURVZ SORC1060
C ACCEL=(BURVZ-VZSPH)/TBURST SORC1070
C VOLSPH=(4./3.)*3.141593*R0**3 SORC1080
C TN0T=T0+DELT SORC1090
C CLAM=DD/W3 SORC1100
C
C CALCULATE CRATER RADIUS AND DEPTH SORC1110
C
C ONCE=.FALSE. SORC1120
C IF(NSUIL.EQ.1)IDX=4 SORC1130
C IF(NSOIL.EQ.2.)IDX=3 SORC1140
C GO TO 70 SORC1150
60 IF(NSOIL.EQ.1)IDX=6 SORC1160
70 IF(NSOIL.EQ.2)IDX=4 SORC1170
CONTINUE SORC1180
RC=CR(1,IDX) SORC1190
DC=CD(1,IDX) SORC1200
IF(CLAM.LT.1.E-30) GO TO 98 SORC1210
TERM=1. SORC1220
DO 100 I=2,5 SORC1230
TERM=TERM*CLAM SORC1240
RC=RC + CR(I,IDX)*TERM SORC1250
DC=DC + CD(I,IDX)*TERM SORC1260

```

```

100  CONTINUE SORC1410
98   CONTINUE SORC1420
      RC=RC*W3 SORC1430
      DC=DC*(W*WTRAT(NCHRG))**.3 SORC1440
      SORC1450
      SORC1460
      SORC1470
C GET CRATER VOLUME SORC1480
      DSDD=DSOD/DC SORC1490
      VC=(2.*3.141592/3.)*RC*RC*DC*(1.-1.5*DSDD*(1.-DSDD*DSDC/3.)) SORC1500
      IF(DSOD.GE.DC)VC=0.0 SORC1510
      IF(ONCE)GO TO 110 SORC1520
      ONCE=.TRUE. SORC1530
      IF(NSOIL.EQ.1)VC1=.5*VC SORC1540
      IF(NSOIL.EQ.2)VC1=.25*VC SORC1550
      GO TO 60 SORC1560
110  IF(NSOIL.EQ.1)VC=VC1+.5*VC SORC1570
      IF(NSOIL.EQ.2)VC=VC1+.75*VC SORC1580
C CALCULATE OPTICALLY WEIGHTED PARAMETERS SORC1590
      NDSCS=MIN0(10,IFIX(5.*W3/1.8))
      CBLEED=0. SORC1600
      DO 101 L=1,NPRTS SORC1620
      S(L)=OWML(L,NSOIL)*VC SORC1630
      VGRAV(L)=OWSY(L,NSOIL) SORC1640
      SPHNS(L)=PRTTN(NSOIL)*S(L) SORC1650
      QDSC(1,L)=(1.-PRTTN(NSOIL))*S(L)/FLOAT(NDSCS) SORC1660
      CBLEED=CBLEED+S(L) SORC1670
101  CONTINUE SORC1680
      CBLEED=CBLEED*.03/W3**3 SORC1690
      RCARB=SCARB/SPHNS(1) SORC1700
      RCARB1=1./(1.+RCARB) SORC1710
      RCARB2=RCARB/(1.+RCARB) SORC1720
      SPHNS(1)=SPHNS(1)+SCARB SORC1730
      DELH=2.*R0/FLOAT(NDSCS) SORC1740
      Z=-DELH/2. SORC1750
      DO 200 I=1,NDSCS SORC1760
      Z=Z+DELH SORC1770
      ZDSC(I)=Z SORC1780
      DO 201 J=1,NPRTS SORC1790
      QDSC(I,J)=QDSC(1,J) SORC1800
201  CONTINUE SORC1810
      CON=ALOG(QDSC(I,1)/VISEXT/DELH/(2.*R0)/3.14159) SORC1820
      IF(CON.GT.1.)GO TO 210 SORC1830
      D=1. SORC1840
      GO TO 230 SORC1850
210  D=CON SORC1860
      DO 220 IT=1,5 SORC1880
      D=(CON-1.+ALOG(D))*D/(D-1) SORC1890
220  CONTINUE SORC1900
230  R2DSC(I)=4.*R0*R0/D SORC1910
      TDSC(I)=-DELH*DELH/D/(DZ0*Z**DN)/4. SORC1920
      SIGZ=DELH*DELH/D SORC1930
      XDSC(I)=U0*Z**UM * TDSC(I) SORC1940
200  CONTINUE SORC1950
999   RETURN SORC1960
      END SORC1970

```

```

SUBROUTINE TEMP(Z,TA,DTADZ)                               TEMP0240
C PURPOSE                                                 TEMP0010
  TO COMPUTE THE AMBIENT ATMOSPHERIC POTENTIAL TEMPERATURE AND TEMP0020
  GRADIENT AT A GIVEN HEIGHT.                           TEMP0030
C INPUTS                                                 TEMP0040
  Z  HEIGHT AT WHICH AMBIENT TEMPERATURE AND TEMPERATURE TEMP0050
  GRADIENT ARE DESIRED.                                TEMP0060
C OUTPUTS                                                TEMP0070
  TA  AMBIENT POTENTIAL TEMPERATURE                      TEMP0080
  DTADZ TEMPERATURE GRADIENT                            TEMP0090
C SUBROUTINES AND FUNCTIONS NEEDED                     TEMP0100
  TMPCAL COMPUTES SCALED TEMPERATURE AT A GIVEN HEIGHT TEMP0110
C CALLED BY DIFEQ, ATMCAL                             TEMP0120
C *****
COMMON/STARS/USTAR,TSTAR,ZSTAR                         TEMP0250
COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT                 TEMP0260
COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT                  TEMP0270
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                   TEMP0280
COMMON/IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU TEMP0290
S=Z/ZL
TA=TSTAR*TMPCAL(Z0,ZL,Z)+T0                          TEMP0300
IF(ABS(ZL).LE.1.E03)GO TO 10                         TEMP0310
C NEUTRAL CASE                                         TEMP0320
  DTADZ=TSTAR/(Z0+Z)                                    TEMP0330
  GO TO 999
  10 IF(ZL.GT.0.0)GO TO 15                           TEMP0340
C UNSTABLE CASE                                         TEMP0350
  DTADZ=(TSTAR/Z)*(1.-16.*S)**(-1./2.)               TEMP0360
  IF(S.LT.-2.0)DTADZ=(TSTAR/ZL)*(AT/3.*(-ZL/Z)**(4./3.)) TEMP0370
  GO TO 999
C STABLE CASE                                           TEMP0380
  15 DTADZ=(TSTAR/ZL)*(ZL/(Z0+Z)+11.)                TEMP0390
  IF(S.GT.1.5)DTADZ=BT*TSTAR/ZL                        TEMP0400
  999 RETURN                                            TEMP0410
  END

```

```

FUNCTION TMPCAL(Z0,ZL,Z)
***** TMP00010
TMP00020
TMP00030
TMP00040
TMP00050
TMP00060
TMP00070
TMP00080
TMP00090
TMP00100
TMP00110
TMP00120
TMP00130
TMP00140
TMP00150
TMP00160
TMP00170
TMP00180
TMP00190
TMP00200
TMP00210
TMP00220
TMP00230
TMP00240
TMP00250
TMP00260
***** TMP00270
TMP00280
TMP00290
TMP00300
TMP00310
TMP00320
TMP00330
TMP00340
TMP00350
TMP00360
TMP00370
TMP00380
TMP00390
TMP00400
TMP00410
TMP00420
TMP00430
TMP00440
TMP00450
TMP00460
TMP00470
TMP00480
TMP00490
TMP00500
TMP00510
TMP00520
TMP00530
TMP00540
TMP00550
TMP00560
TMP00570
TMP00580
TMP00590
TMP00600
TMP00610
TMP00620
TMP00630
TMP00640
TMP00650
TMP00660
TMP00670
TMP00680
TMP00690
TMP00700
C PURPOSE
TO CALCULATE THE POTENTIAL TEMPERATURE SCALED BY THE SCALE
TEMPERATURE, T*, FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV
LENGTH AT A SPECIFIED HEIGHT.
C INPUTS
Z0      THE FRICTION HEIGHT IN METERS.
ZL      THE MONIN-OBUKHOV LENGTH IN METERS.
Z       THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
IN METERS
C RETURNS SCALED TEMPERATURE
C CALLED BY ATMCAL, SOURCE AND TEMP
C **** LOGICAL LOW
COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
COMMON/IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIR TU,NCLIM,T,KSTOR,NPLOTU
PHIM(Z)=1.-16.*Z**(-.25)
PSIH(S,S0)= ALOG(((S**2.-1.)/(S**2.+1.))*((S0**2.+1.)/(S0**2.-1.)))
PSIHS(Z)=-11.*Z
TRACE 999
C
PHIM      THE SHEAR OF MOMENTUM
PSIH      THE UNIVERSAL FUNCTION FOR DEVIATION FROM LOGARITHMIC
          POTENTIAL TEMPERATURE PROFILE IN THE BOUNDARY LAYER
          OF AN UNSTABLE ATMOSPHERE
PHIHS     THE SAME AS PHIHS EXCEPT FOR STABLE ATMOSPHERE
IF(Abs(ZL).LE.1.E3)GO TO 100
TMPCAL=ALOG(1.+Z/Z0)
GO TO 999
100 CONTINUE
P=SIGN(1.,ZL)
LOW=.TRUE.
S=Z/ZL
IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
S=AMIN1(S,1.5)
S=AMAX1(S,-2.)
LOW=.FALSE.
10 CONTINUE
IF(P>120,130,130
120 S=1./PHIM(S)
S1=Z0/ZL
S0=1./PHIM(S1)
TMPCAL=PSIH(S,S0)
C FIND CONSTANTS FOR MATCHING IN UNSTABLE CASE AT Z/ZL=-2.
S2=-2.
AT=-3.*((1.-16.*S2)**(-1./2.))*(-S2)**(1./3.)
CT=-1.*AT*(-S2)**(-1./3.)
GO TO 52
130 CONTINUE
PSI=PSIHS(S)
TMPCAL=ALOG(1.+S*ZL/Z0)-PSI
C FIND CONSTANTS FOR MATCHING IN STABLE CASE AT Z/ZL=1.5

```

C
S2=1.5
BT=1./((Z0/ZL+S2)+11.
DT=-1.*BT*S2
52 CONTINUE
IF(LOW)GO TO 999
IF(P>53,53,54
53 TMPCAL=TMPCAL+CT+AT*(-ZL/Z)**(1./3.)
GO TO 999
54 TMPCAL=TMPCAL+DT+BT*Z/ZL
999 RETURN
END

TMP00710
TMP00720
TMP00730
TMP00740
TMP00750
TMP00760
TMP00770
TMP00780
TMP00790
TMP00800
TMP00810
TMP00820

SUBROUTINE TRAP(TRSK,TRWK,TRSP,H,SIGW,SIGO,TIME,SKT,WAK,SPH) TRP00510
 THIS SUBROUTINE PERFORMS A TRAPEZOID INTEGRATION TRP00010
 INPUTS TRP00020
 TRP00030
 TRP00040
 TRP00050
 TRP00060
 TRP00070
 TRP00080
 TRP00090
 TRP00100
 TRP00110
 TRP00120
 TRP00130
 TRP00140
 TRP00150
 TRP00160
 TRP00170
 TRP00180
 TRP00190
 TRP00200
 TRP00210
 TRP00220
 TRP00230
 TRP00240
 TRP00250
 TRP00260
 TRP00270
 TRP00280
 TRP00290
 TRP00300
 TRP00310
 TRP00320
 TRP00330
 TRP00340
 TRP00350
 TRP00360
 TRP00370
 TRP00380
 TRP00390
 TRP00400
 TRP00410
 TRP00420
 TRP00430
 TRP00440
 TRP00450
 TRP00460
 TRP00470
 TRP00480
 TRP00490
 TRP00500
 TRP00520
 TRP00530
 TRP00540
 TRP00550
 TRP00560
 TRP00570
 TRP00580
 TRP00590
 TRP00600
 TRP00610
 TRP00620
 TRP00630
 TRP00640
 TRP00650
 TRP00660
 TRP00670
 TRP00680
 TRP00690
 TRP00700

SKT - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT FROM THE SKIRT TRP00270
 WAK - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT FROM THE WAKE ONCE THE BUOYANT SPHERE HAS CONVERTED TO THE WIND MODEL TRP00310
 SPH - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT FROM THE SPHERE ONCE IT HAS CONVERTED TO THE WIND MODEL TRP00340

FUNCTIONS NEEDED

CWIND - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SKIRT TRP00410
 CWAKE - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE WAKE TRP00420
 CSPHER - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SPHERE TRP00430

```

REAL KX,KZ
LOGICAL SWITCH,CHANGE,ONCE
DIMENSION OWF(5,2),OWFC(5),TRSK(3),TRWK(3),TRSP(3)
COMMON/LOS/TR(3),RE(3),U(3)
COMMON/ACL/CWINDS,CWINDC,CWINDW
COMMON/CARB/RCARB1,RCARB2
COMMON/TRAN/VTR,K2,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON/POINTS/XNORM,DOT1,DOT2,DOT3
DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,4.E-03/
DATA OWFC/1.,.95,.5,.2,1.E-03/
TI=TIME
SKT=0.0
WAK=0.0
SPH=0.0
SUM=0.0
SUM1=0.0
SUM2=0.0
IND=20
  
```

```

ONCE=.FALSE.
FP=0.0
F=0.0
J=0
DO 10 I=1,IND
XI=H*FLOAT(I-1)
DIST=D0T1+XI
IF(DIST.GT.XNORM)GO TO 10
IF(DIST.LT.0.0)GO TO 10
X=TRSK(1)+XI*U(1)
Y=TRSK(2)+XI*U(2)
Z=TRSK(3)+XI*U(3)
FP=F
ACL=CWIND(X,Y,Z,TI)
F=ACL
      TRP00710
      TRP00720
      TRP00730
      TRP00740
      TRP00750
      TRP00760
      TRP00770
      TRP00780
      TRP00790
      TRP00800
      TRP00810
      TRP00820
      TRP00830
      TRP00840
      TRP00850
      TRP00860
      TRP00870
      TRP00880
      TRP00890
      TRP00900
      TRP00910
      TRP00920
      TRP00930
      TRP00940
      TRP00950
      TRP00960
      TRP00970
      TRP00980
      TRP00990
      TRP01000
      TRP01010
      TRP01020
      TRP01030
      TRP01040
      TRP01050
      TRP01060
      TRP01070
      TRP01080
      TRP01090
      TRP01100
      TRP01110
      TRP01120
      TRP01130
      TRP01140
      TRP01150
      TRP01160
      TRP01170
      TRP01180
      TRP01190
      TRP01200
      TRP01210
      TRP01220
      TRP01230
      TRP01240
      TRP01250
      TRP01260
      TRP01270
      TRP01280
      TRP01290
      TRP01300
      TRP01310
      TRP01320
      TRP01330
      TRP01340
      TRP01350
      TRP01360
      TRP01370
      TRP01380
      TRP01390
      TRP01400

C CHECK TO SEE IF THE CONTRIBUTION IS NEGLIGABLE
      TRP00820
      TRP00830
      TRP00840
      TRP00850
      TRP00860
      TRP00870
      TRP00880
      TRP00890
      TRP00900
      TRP00910
      TRP00920
      TRP00930
      TRP00940
      TRP00950
      TRP00960
      TRP00970
      TRP00980
      TRP00990
      TRP01000
      TRP01010
      TRP01020
      TRP01030
      TRP01040
      TRP01050
      TRP01060
      TRP01070
      TRP01080
      TRP01090
      TRP01100
      TRP01110
      TRP01120
      TRP01130
      TRP01140
      TRP01150
      TRP01160
      TRP01170
      TRP01180
      TRP01190
      TRP01200
      TRP01210
      TRP01220
      TRP01230
      TRP01240
      TRP01250
      TRP01260
      TRP01270
      TRP01280
      TRP01290
      TRP01300
      TRP01310
      TRP01320
      TRP01330
      TRP01340
      TRP01350
      TRP01360
      TRP01370
      TRP01380
      TRP01390
      TRP01400

C COMPUTE THE CONTRIBUTION TO CL FROM THE SPHERE USING A TRAPEZOID
C INTEGRATION ONLY AFTER THE BUOYANT FIREBALL HAS CONVERTED TO THE WIND
C MODEL
      TRP01000
      TRP01010
      TRP01020
      TRP01030
      TRP01040
      TRP01050
      TRP01060
      TRP01070
      TRP01080
      TRP01090
      TRP01100
      TRP01110
      TRP01120
      TRP01130
      TRP01140
      TRP01150
      TRP01160
      TRP01170
      TRP01180
      TRP01190
      TRP01200
      TRP01210
      TRP01220
      TRP01230
      TRP01240
      TRP01250
      TRP01260
      TRP01270
      TRP01280
      TRP01290
      TRP01300
      TRP01310
      TRP01320
      TRP01330
      TRP01340
      TRP01350
      TRP01360
      TRP01370
      TRP01380
      TRP01390
      TRP01400

C COMPUTE CONTRIBUTION TO CL FROM THE WAKE AFTER THE BUOYANT FIREBALL
C HAS CONVERTED TO THE WIND MODEL USING TRAPEZOID INTEGRATION WITH STEP
C SIZE SIGW.
      TRP01300
      TRP01310
      TRP01320
      TRP01330
      TRP01340
      TRP01350
      TRP01360
      TRP01370
      TRP01380
      TRP01390
      TRP01400

C ONCE=.FALSE.
FP=0.0
F=0.0
J=0
DO 30 I=1,IND
XI=SIGW*FLOAT(I-1)
      TRP01300
      TRP01310
      TRP01320
      TRP01330
      TRP01340
      TRP01350
      TRP01360
      TRP01370
      TRP01380
      TRP01390
      TRP01400

```

```

DIST=DOT2+XI           TRP01410
IF(DIST.GT.XNORM)GO TO 30   TRP01420
IF(DIST.LT.0.0)GO TO 30   TRP01430
X=TRWK(1)+XI*U(1)        TRP01440
Y=TRWK(2)+XI*U(2)        TRP01450
Z=TRWK(3)+XI*U(3)        TRP01460
FP=F                      TRP01470
ACL=CWAKE(X,Y,Z,TI)      TRP01480
F=ACL                     TRP01490
IF(.NOT.ONCE)ACL1=ACL    TRP01500
IF(ACL1.LT.1.E-05)GO TO 31   TRP01510
PER=.01*ACL1              TRP01520
ONCE=.TRUE.                TRP01530
J=J+1                     TRP01540
IF(ACL.LT.PER)GO TO 31    TRP01550
IF(J.LE.1)GO TO 30         TRP01560
SUM2=SUM2+FP+F             TRP01570
30 CONTINUE                 TRP01580
31 CONTINUE                 TRP01590
SKT=(ABS(H)/2.)*SUM         TRP01600
WAK=(ABS(SIGN)/2.)*SUM2     TRP01610
SPH=(ABS(SIG0)/2.)*SUM1     TRP01620
999 RETURN                  TRP01630
END                         TRP01640

```

SUBROUTINE TRNCAL(TRN,REC,TIME,TRNLOS)	TRL00520
CONTROLING ROUTINE FOR CALCULATING TRANSMITTANCES FOR CHARGE	TRL00010
DISTRIBUTION TYPES 1 AND 2	TRL00020
INPUTS	TRL00030
TRN - TRANSMITTER COORDINATES IN THE LOCAL COORDINATE SYSTEM	TRL00040
REC - RECEIVER COORDINATES IN THE LOCAL COORDINATE SYSTEM	TRL00050
TIME - TIME AFTER THE DETONATION AT WHICH A TRANSMITTANCE IS DESIRED	TRL00060
ALL OTHER NECESSARY INPUTS ARE PASSED IN COMMON BLOCKS	TRL00070
OUTPUT	TRL00080
TRNLOS - TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT	TRL00090
SUBROUTINES NEEDED	TRL00100
AVRG - FINDS THE AVERAGE OF THE MOMENTS FOR THE DISCS	TRL00110
VSUM - ADDS TWO VECTORS	TRL00120
UNIT - DETERMINE A UNIT VECTOR	TRL00130
TRNCLD - DETERMINE THE LENGTH OF THE INTERSECTION OF THE LINE OF	TRL00140
SIGHT WITH THE WAKE AND SPHERE	TRL00150
TRAP - DOES A TRAPEZOIDAL INTEGRATION THROUGH SKIRT WAKE AND S	TRL00160
SPHERE FOR NON-HORIZONTAL LINES OF SIGHT	TRL00170
TRNCHK - CHECKS TO SEE IF THE OBSCURATION IS SUCH THAT THE	TRL00180
TRANSMITTANCE IS LESS THAN A SPECIFIED VALUE	TRL00190
FUNCTIONS NEEDED	TRL00200
DOTPRD - FINDS THE DOTPRODUCT OF TWO VECTORS	TRL00210
CWIND - FINDS THE CONCENTRATION ALONG A SPECIFIED HORIZONTAL	TRL00220
LINE OF SIGHT OR DETERMINES THE CONCENTRATION AT SOME	TRL00230
POINT ALONG THE LINE OF SIGHT FROM THE SKIRT	TRL00240
CWAKE - SAME AS CWIND EXCEPT FOR WAKE	TRL00250
CSPHER - SAME AS CWIND EXCEPT FOR BUOYANT SPHERE	TRL00260
*****	TRL00270
REAL KZ,KX	TRL00280
DIMENSION TRN(3),REC(3),OWF(5,2),OWFC(5),TEMP(2)	TRL00290
DIMENSION DIR(2),	TRL00300
XW(3),XS(3)	TRL00310
DIMENSION TRSK(3),TRWK(3),TRSP(3)	TRL00320
LOGICAL HORIZ,SWITCH,CHANGE,TEST,SKIP	TRL00330
COMMON /IOUNIT/IOUT,IOIN,IOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUT	TRL00340
COMMON/CARB/RCARB1,RCARB2	TRL00350
COMMON/BUOYCL/RSPH,DELT,YZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM	TRL00360
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,	TRL00370
+ DMM(600),DMMYC(401)	TRL00380
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)	TRL00390
COMMON/MODE/HORIZ	TRL00400
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE	TRL00410
COMMON/ACL/CWINDS,CWINDC,CWINDW	TRL00420
COMMON/LOS/T(3),R(3),U(3)	TRL00430
COMMON/SIG/SIG02,SIGC	TRL00440
COMMON/CHARGE/NCHG	TRL00450
COMMON/POINTS/XNORM,DOT1,DOT2,DOT3	TRL00460

```

COMMON/TRANNS/THRESH, TEST, NWL, NSOIL TRL 00710
COMMON/SKIPIT/SKIP TRL 00720
COMMON/EKTEMP/20, ZL, T0, TC1, TC2, TC3 TRL 00730
DATA DNEM/-1.0/ TRL 00740
DATA QWF/1., .93, .52, .44, 2.E-03, 1., 1., 1., 4.E-03/ TRL 00750
DATA QWFC/1., .95, .5, .2, 1.E-03/ TRL 00760
C PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER TRL 00770
C
NCHG=1 TRL 00780
SKIP=.FALSE. TRL 00790
TEST=.FALSE. TRL 00800
HSPH=0.0 TRL 00810
HWAK=0.0 TRL 00820
ACLSKT=0.0 TRL 00830
ACLMWK=0.0 TRL 00840
ACLSPH=0.0 TRL 00850
XNORM=0.0 TRL 00860
DO 10 I=1,3 TRL 00870
R(I)=REC(I) TRL 00880
T(I)=TRN(I) TRL 00890
U(I)=R(I)-T(I) TRL 00900
XNORM=XNORM+U(I)**2 TRL 00910
10 CONTINUE TRL 00920
XNORM=SQRT(XNORM) TRL 00930
U(1)=U(1)/XNORM TRL 00940
U(2)=U(2)/XNORM TRL 00950
U(3)=U(3)/XNORM TRL 00960
IF((TIME-TTR).LT.1.E-20)GO TO 14 TRL 00970
C IF THE BUOYANT SPHERE HAS BECOME WIND BLOWN DETERMINE THE CENTER OF TRL 00980
C MASS OF THE REFERENCE CHARGE. TRL 00990
C XCM=XTR+VTR*(TIME-TTR) TRL 01000
ZCM=ZTR TRL 01010
C DETERMINE CENTER OF WAKE FOR REFERENCE CHARGE TRL 01020
C
14 ZX=5.0 TRL 01030
CALL AYRG(ZX, TIME, QTOT, XBAVRG, SIG2X, SIG2Y) TRL 01040
XW(1)=(XBAVRG+XCM)/2. TRL 01050
XW(2)=YCM/2. TRL 01060
XW(3)=(5.0+ZCM)/2. TRL 01070
C IF THE DIFFERENCE BETWEEN THE TRANSMITTER AND RECEIVER IS GREATER TRL 01080
C THAN 1 PERCENT OF THE DISTANCE BETWEEN THEM THEN THE LOS IS TRL 01090
C CONSIDERED A SLANT PATH. TRL 01100
C
13 IF(ABS(U(3)).GT..01)GO TO 20 TRL 01110
IF(ABS(TRN(3)-REC(3)).LT.1.E-06)GO TO 9 TRL 01120
C COMPUTE CLOSEST POINT ALONG THE LOS TO OUR ESTIMATE OF THE CENTER OF TRL 01130
C THE WAKE TRL 01140
C
DDT=-U(1)*(T(1)-XW(1))+U(2)*(T(2)-XW(2))+U(3)*(T(3)-XW(3)) TRL 01150
T(3)=T(3)+U(3)*DDT TRL 01160
R(3)=T(3) TRL 01170
C COMPUTE CONTRIBUTIONS FOR A HORIZONTAL PATH TRL 01180
C
9 HORIZ=.TRUE. TRL 01190
CALL VSUM(REC, TRN, DNEM, DIR) TRL 01200
CALL UNIT(DIR, DIR, RANGE) TRL 01210
COSTH=DIR(1) TRL 01220
SINTH=DIR(2) TRL 01230
SINTH2=SINTH*SINTH TRL 01240
COSTH2=COSTH**2 TRL 01250
SCRN(1)=SINTH TRL 01260
SCRN(2)=-COSTH TRL 01270
DO 12 J=1, ITOT TRL 01280

```

```

DO 11 I=1,2
  TEMP(I)=DIFF(I,J)
11 CONTINUE
  PRSEP(J)=DOTPRD(TEMP,SCRN)
12 CONTINUE
  X=DOTPRD(SCRN,TRN)
      TRL01410
      TRL01420
      TRL01430
      TRL01440
      TRL01450
      TRL01460
      TRL01470
      TRL01480
      TRL01490
      TRL01500
      TRL01510
      TRL01520
      TRL01530
      TRL01540
      TRL01550
      TRL01560
      TRL01570
      TRL01580
      TRL01590
      TRL01600
      TRL01610
      TRL01620
      TRL01630
      TRL01640
      TRL01650
      TRL01660
      TRL01670
      TRL01680
      TRL01690
      TRL01700
      TRL01710
      TRL01720
      TRL01730
      TRL01740
      TRL01750
      TRL01760
      TRL01770
      TRL01780
      TRL01790
      TRL01800
      TRL01810
      TRL01820
      TRL01830
      TRL01840
      TRL01850
      TRL01860
      TRL01870
      TRL01880
      TRL01890
      TRL01900
      TRL01910
      TRL01920
      TRL01930
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C COMPUTE THE CONTRIBUTION FROM THE SKIRT AT A HEIGHT OF T(3) WHERE T(3) IS THE HEIGHT OF THE TRANSMITTER IF THE DIFFERENCE BETWEEN THE TRANSMITTER AND RECEIVER HEIGHTS IS SMALL AND IS THE Z COMPONENT OF THE POINT ON THE LINE CONNECTING THE TRANSMITTER AND RECEIVER WHICH IS CLOSEST TO OUR ESTIMATE OF THE CENTER OF THE WAKE OTHERWISE.
      TRL01480
      TRL01490
      TRL01500
      TRL01510
      TRL01520
      TRL01530
      TRL01540
      TRL01550
      TRL01560
      TRL01570
      TRL01580
      TRL01590
      TRL01600
      TRL01610
      TRL01620
      TRL01630
      TRL01640
      TRL01650
      TRL01660
      TRL01670
      TRL01680
      TRL01690
      TRL01700
      TRL01710
      TRL01720
      TRL01730
      TRL01740
      TRL01750
      TRL01760
      TRL01770
      TRL01780
      TRL01790
      TRL01800
      TRL01810
      TRL01820
      TRL01830
      TRL01840
      TRL01850
      TRL01860
      TRL01870
      TRL01880
      TRL01890
      TRL01900
      TRL01910
      TRL01920
      TRL01930
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C ACLSKT=CWIND(X,Y,T(3),TIME)
  IF(TEST)GO TO 998
      TRL01530
      TRL01540
      TRL01550
      TRL01560
      TRL01570
      TRL01580
      TRL01590
      TRL01600
      TRL01610
      TRL01620
      TRL01630
      TRL01640
      TRL01650
      TRL01660
      TRL01670
      TRL01680
      TRL01690
      TRL01700
      TRL01710
      TRL01720
      TRL01730
      TRL01740
      TRL01750
      TRL01760
      TRL01770
      TRL01780
      TRL01790
      TRL01800
      TRL01810
      TRL01820
      TRL01830
      TRL01840
      TRL01850
      TRL01860
      TRL01870
      TRL01880
      TRL01890
      TRL01900
      TRL01910
      TRL01920
      TRL01930
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C TEST IS A LOGICAL VARIABLE RETURNED IN COMMON/TRANNY/ FROM SUBROUTINE TRNCHK WHICH IS CALLED BY CWIND, CWAKE, CSPHER, TRNCLD, AND TRAF EACH TIME A CONTRIBUTION IS MADE TO THE OPTICALLY WEIGHTED CONCENTRATION ALONG THE OPTICAL PATH
      TRL01530
      TRL01540
      TRL01550
      TRL01560
      TRL01570
      TRL01580
      TRL01590
      TRL01600
      TRL01610
      TRL01620
      TRL01630
      TRL01640
      TRL01650
      TRL01660
      TRL01670
      TRL01680
      TRL01690
      TRL01700
      TRL01710
      TRL01720
      TRL01730
      TRL01740
      TRL01750
      TRL01760
      TRL01770
      TRL01780
      TRL01790
      TRL01800
      TRL01810
      TRL01820
      TRL01830
      TRL01840
      TRL01850
      TRL01860
      TRL01870
      TRL01880
      TRL01890
      TRL01900
      TRL01910
      TRL01920
      TRL01930
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C TEST=.FALSE. TRANSMITTANCE IS GREATER THAN TRNMIN
  (A TRANSMITTANCE THRESHOLD).
  =.TRUE. TRANSMITTANCE IS LESS THAN TRNMIN
      TRL01630
      TRL01640
      TRL01650
      TRL01660
      TRL01670
      TRL01680
      TRL01690
      TRL01700
      TRL01710
      TRL01720
      TRL01730
      TRL01740
      TRL01750
      TRL01760
      TRL01770
      TRL01780
      TRL01790
      TRL01800
      TRL01810
      TRL01820
      TRL01830
      TRL01840
      TRL01850
      TRL01860
      TRL01870
      TRL01880
      TRL01890
      TRL01900
      TRL01910
      TRL01920
      TRL01930
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C IF(TIME.GT.TTR)ACLWAK=CWAKE(X,Y,T(3),TIME)
  IF(TEST)GO TO 998
  IF(TIME.GT.TTR)ACLSPH=CSPHER(X,Y,T(3),TIME)
  IF(TEST)GO TO 998
  IF(TIME.GT.TTR)GO TO 50
  CWINDS=ACLWAK
  CWINDM=ACLWAK
  CWINDC=ACLSPH
  CALL TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)
  IF(TEST)GO TO 998
  GO TO 50
      TRL01630
      TRL01640
      TRL01650
      TRL01660
      TRL01670
      TRL01680
      TRL01690
      TRL01700
      TRL01710
      TRL01720
      TRL01730
      TRL01740
      TRL01750
      TRL01760
      TRL01770
      TRL01780
      TRL01790
      TRL01800
      TRL01810
      TRL01820
      TRL01830
      TRL01840
      TRL01850
      TRL01860
      TRL01870
      TRL01880
      TRL01890
      TRL01900
      TRL01910
      TRL01920
      TRL01930
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C DO TRAPEZOIDAL INTEGRATION FOR SLANT PATH IN BOTH DIRECTIONS FROM AN ESTIMATE OF THE LOCATION OF CENTER OF THE SKIRT USING A STEP SIZE OF SIG, THE GEOMETRIC MEAN OF THE AVERAGE OF THE SPREADS OF THE DISCS IN BOTH THE X AND Y DIRECTION. THEN IF THE BUOYANT SPHERE HAS CONVERTED TO THE WIND MODEL DO THE SAME FOR THE WAKE AND SPHERE WITH THE APPROPRIATE STEP SIZE. (CHECK TO SEE IF SPHERE HAS CONVERTED TO THE WIND MODEL IS DONE IN TRAP.)
      TRL01800
      TRL01810
      TRL01820
      TRL01830
      TRL01840
      TRL01850
      TRL01860
      TRL01870
      TRL01880
      TRL01890
      TRL01900
      TRL01910
      TRL01920
      TRL01930
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C 20 HOFIC=.FALSE.
  SIGX=SQRT(SIG2X)
  SIGY=SQRT(SIG2Y)
  SIG=SQRT(SIGX*SIGY)
  CALL MIN(2.0,UX,UY)
  XSC(1)=TIME*UX
  XSC(2)=TIME*UY
  XSC(3)=2.0
      TRL01860
      TRL01870
      TRL01880
      TRL01890
      TRL01900
      TRL01910
      TRL01920
      TRL01930
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C FIND THE POINTS ON THE LINE CONNECTING THE TRANSMITTER AND RECEIVER THAT ARE CLOSEST TO OUR ESTIMATE OF THE CENTER OF THE SKIRT, WAKE, AND SPHERE.
      TRL01940
      TRL01950
      TRL01960
      TRL01970
      TRL01980
      TRL01990
      TRL02000
      TRL02010
      TRL02020
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

C DO 48 J=1,ITOT
  XSK=XSC(1)+DIFF(1,J)
  YSK=XSC(2)+DIFF(2,J)
  ZSK=XSC(3)
  DOT1=-U(1)*(T(1)-XSK)+U(2)*(T(2)-YSK)+U(3)*(T(3)-ZSK)
  IF(DOT1.LT.0.0)DOT1=0.0
  IF(DOT1.GT.XNORM)DOT1=XNORM
  XWK=XSC(1)+DIFF(1,J)
  YWK=XSC(2)+DIFF(2,J)
  ZWK=XSC(3)
  DOT2=-U(1)*(T(1)-XWK)+U(2)*(T(2)-YWK)+U(3)*(T(3)-ZWK)
  IF(DOT2.LT.0.0)DOT2=0.0
      TRL02030
      TRL02040
      TRL02050
      TRL02060
      TRL02070
      TRL02080
      TRL02090
      TRL02100

```

```

IF(DOT2.GT.XNORM)DOT2=XNORM
XSP=XCM+DIFF(1,J)
YSP=YCM+DIFF(2,J)
ZSP=ZCM
DOT3=-U(1)*(T(1)-XSP)+U(2)*(T(2)-YSP)+U(3)*(T(3)-ZSP)
IF(DOT3.LT.0.0)DOT3=0.0
IF(DOT3.GT.XNORM)DOT3=XNORM
NCHG=J
DO 45 II=1,3
TRSK(II)=T(II)+U(II)*DOT1
TRWK(II)=T(II)+U(II)*DOT2
TRSP(II)=T(II)+U(II)*DOT3
45 CONTINUE
DIFS=DIFFUS(Z0,ZL,2,0)
SIGZ=SQRT(2.*DIFS*TIME)
IF(SIGZ.LT.1.0)SIGZ=1.0
H=U(1)*SIGX+U(2)*SIGY+U(3)*SIGZ
IF(TIME.LT.TTR)GO TO 46
HSPX=SQRT(SIGZ*2.*KX*(TIME-TTR))
HSPY=HSPX
HSPZ=SQRT(SIGZ*2.*KZ*(TIME-TTR))
HSPH=(U(1)*HSPX+U(2)*HSPY+U(3)*HSPZ)/2.
HWAK=(HSPH+H)/2.
46 CONTINUE
CALL TRAP(TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH)
ACLSKT=ACLSKT+SKT
ACLWAK=ACLWAK+WAK
ACLSPH=ACLSPH+SPH
CALL TRNCHK(ACLSKT,ACLWAK,ACLSPH)
IF(TEST)GO TO 998
H=-H
IF(TIME.LT.TTR)GO TO 47
HSPH=-HSPH
HWAK=-HWAK
47 CONTINUE
CALL TRAP(TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH)
ACLSKT=ACLSKT+SKT
ACLWAK=ACLWAK+WAK
ACLSPH=ACLSPH+SPH
CWINDW=ACLSKT
CWINDS=ACLSPH
CALL TRNCHK(ACLSKT,ACLWAK,ACLSPH)
IF(TEST)GO TO 998
48 CONTINUE
IF(TIME.GT.TTR)GO TO 50
CALL TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)
IF(TEST)GO TO 998
50 CONTINUE
ACLC=(ACLWAK+ACLSPH)*(RCARB1*DWF(NWL,NSOIL)+RCARB2*DWF(NWL))
ACLS=ACLSKT*DWF(NWL,NSOIL)
C COMPUTE THE TRANSMITTANCE ALONG THE LINE OF SIGHT
TRNLDS=EXP(-ACLS-ACLC)
GO TO 999
998 TRNLDS=0.0
999 RETURN
END

```

TRL 02110
TRL 02120
TRL 02130
TRL 02140
TRL 02150
TRL 02160
TRL 02170
TRL 02180
TRL 02190
TRL 02200
TRL 02210
TRL 02220
TRL 02230
TRL 02240
TRL 02250
TRL 02260
TRL 02270
TRL 02280
TRL 02290
TRL 02300
TRL 02310
TRL 02320
TRL 02330
TRL 02340
TRL 02350
TRL 02360
TRL 02370
TRL 02380
TRL 02390
TRL 02400
TRL 02410
TRL 02420
TRL 02430
TRL 02440
TRL 02450
TRL 02460
TRL 02470
TRL 02480
TRL 02490
TRL 02500
TRL 02510
TRL 02520
TRL 02530
TRL 02540
TRL 02550
TRL 02560
TRL 02570
TRL 02580
TRL 02590
TRL 02600
TRL 02610
TRL 02620
TRL 02630
TRL 02640
TRL 02650
TRL 02660
TRL 02670
TRL 02680
TRL 02690

SUBROUTINE TRNCHK(ACLS,ACLW,ACLC) TRK00310
 THIS IS A SUBROUTINE TO CHECK IF THE CONCENTRATIONS ARE HIGH ENOUGH TRK00010
 SO THAT THE TRANSMITTANCE WILL BE BELOW A GIVEN LEVEL. TRK00020
 INPUTS TRK00030
 ACLS - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT TRK00040
 DUE TO THE SKIRT TRK00050
 ACLW - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT TRK00060
 DUE TO THE WAKE TRK00070
 ACLC - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT TRK00080
 DUE TO THE BUOYANT SPHERE TRK00090
 ALL OTHER NECESSARY INFORMATION IS PASSED VIA COMMON BLOCKS TRK00100
 OUTPUT TRK00110
 TEST - LOGICAL VARIABLE PASSED IN COMMON/TRANNY/ THAT IS .TRUE. TRK00120
 IF THE CONCENTRATION IS SUCH THAT THE TRANSMITTANCE TRK00130
 ALONG THE LINE OF SIGHT WILL BE LESS THAN THE THRESHOLD TRK00140
 VALUE AND IS .FALSE. OTHERWISE TRK00150
 FUNCTIONS AND SUBROUTINE NEEDED TRK00160
 NONE TRK00170
 **** TRK00180
 DIMENSION DWF(5,2),DWF0(5) TRK00190
 LOGICAL TEST TRK00200
 COMMON/CARB/RCARB1,RCARB2 TRK00210
 COMMON/TRANNY/THRESH,TEST,NWL,NSOIL TRK00220
 DATA DWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/ TRK00230
 DATA DWF0/1.,.95,.5,.2,1.E-03/ TRK00240
 TEST=.FALSE. TRK00250
 ACL=(ACLW+ACLC)*(RCARB1*DWF(NWL,NSOIL)+RCARB2*DWF0(NWL)) TRK00260
 ACL=ACL+ACLS*DWF(NWL,NSOIL) TRK00270
 IF(ACL.GT.THRESH)TEST=.TRUE. TRK00280
 999 RETURN TRK00290
 END TRK00300

```

SUBROUTINE TRNCLD(XNORM,TIME,ACLWAK,ACLSPH) TRD00350
C ROUTINE FOR DETERMINING CONTRIBUTION FROM SPHERE AND WAKE TRD00010
C BEFORE THE BUOYANT SPHERE HAS CONVERTED TO THE WIND MODEL. TRD00020
C INPUTS TRD00030
C XNORM -DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER TRD00040
C TIME -TIME AT WHICH TRANSMITTANCE IS DESIRED TRD00050
C OUTPUTS TRD00060
C ACLWAK -CONTRIBUTION FROM WAKE TO OPTICALLY WEIGHTED CONCENTRATION TRD00070
C ALONG GIVEN LINE OF SIGHT TRD00080
C ACLSPH -CONTRIBUTION FROM BUOYANT SPHERE TO OPTICALLY WEIGHTED TRD00090
C CONCENTRATION ALONG GIVEN LINE OF SIGHT TRD00100
C SUBROUTINES NEEDED TRD00110
C AYRG -COMPUTE THE AVERAGE OF THE MOMENTS FOR THE DISCS TRD00120
C WIN -COMPUTE THE WIND SPEED AT A GIVEN HEIGHT TRD00130
C AMOUNT -COMPUTE THE DISTRIBUTION OF THE LOADING BETWEEN THE BUOYANT TRD00140
C SPHERE AND WAKE. TRD00150
C CONLEN -COMPUTE THE LENGTH OF INTERSECTION OF NON-HORIZONTAL LINE TRD00160
C OF SIGHT WITH A CONICAL SHAPED WAKE TRD00170
C TRNCHK -ROUTINE TO CHECK IF THE OBSCURATION IS SUCH THAT THE TRD00180
C TRANSMITTANCE IS LESS THAN A USET SPECIFIED AMOUNT TRD00190
C **** TRD00200
C LOGICAL HORIZ, TEST TRD00210
C DIMENSION CENTER(3) TRD00220
C COMMON/MODE/HORIZ TRD00230
C COMMON/GEOM/COSTH2,SINTH,SINTH2,YISEXT,RTPI,SCRN(2) TRD00240
C COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM TRD00250
C COMMON/PRTINF/R0,YGRAV(3),NPRTS TRD00260
C COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3) TRD00270
C COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT, TRD00280
C + DMM(600),DMMY(401) TRD00290
C COMMON /IOUNIT/IOUT,IOUNIT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUT TRD00300
C COMMON/LOS/TR(3),RE(3),UC(3) TRD00310
C COMMON/TRANNY/THRESH,TEST,NWL,NSOIL TRD00320
C COMMON /ACL/CWINDS,CWINDC,CWINDW TRD00330
C ACLWAK=0.0 TRD00340
C ACLSPH=0.0 TRD00350
C DETERMINE THE RADIUS OF THE BASE OF THE CONE TRD00360
C ZX=5.0 TRD00370
C CALL AYRG(ZX,TIME,QTOT,XBAVG,SIG2X,SIG2Y) TRD00380
C IF(QTOT.LT.1.E-10)GO TO 33 TRD00390
C SIGX=SQRT(SIG2X) TRD00400
C SIGY=SQRT(SIG2Y) TRD00410
C RB=SQRT(SIGX*SIGY) TRD00420
C GO TO 35 TRD00430
C 33 CALL WIN(5.0,UW,V) TRD00440
C XBAVG=UW*TIME TRD00450
C RB=0.0 TRD00460
C 35 VOLWAK=(PI/3.)*(ZCM-5.0)*(RSPH**2+RSPH*RB+RB**2) TRD00470
C VOLSPH=(4./3.)*PI*RSPH**3 TRD00480
C CALL AMOUNT(VOLSPH,WAKAL,SPHAL) TRD00490
C DO 80 J=1,ITOT TRD00500
C IF (VOLWAK.LE.0.0) GO TO 68 TRD00510
C XB=XBAVG+DIFF(1,J) TRD00520

```

```

YB=DIFF(2,J) TRD00730
XC=XCM+DIFF(1,J) TRD00740
YC=YCM+DIFF(2,J) TRD00750
IF(ABS(U(3)).LE..01)GO TO 110 TRD00760
ZBOT=5.0 TRD00770
CALL CONLENKU,TR,ZCM,ZBOT,XC,YC,RSPH,RB,XB,YB,XNORM,PLENWK) TRD00780
ACLW=PLENWK*WAKAL/VOLWAK TRD00790
GO TO 69 TRD00800
68 ACLW=0.0 TRD00810
69 CONTINUE TRD00820
ACLWAK=ACLWAK+ACLW TRD00830
CALL TRNCHK(CWINDS,ACLWAK,ACLSPH) TRD00840
IF(TEST)GO TO 999 TRD00850
C DETERMINE CONTRIBUTION FROM SPHERE FO A SLANT PATH TRD00860
C TRD00870
CENTER(1)=XCM+DIFF(1,J)-TR(1) TRD00890
CENTER(2)=YCM+DIFF(2,J)-TR(2) TRD00900
CENTER(3)=ZCM-TR(3) TRD00910
CLOSE=U(1)*CENTER(1)+U(2)*CENTER(2)+U(3)*CENTER(3) TRD00920
CON=CENTER(1)**2+CENTER(2)**2+CENTER(3)**2-RSPH**2 TRD00930
RADIC=CLOSE**2-CON TRD00940
IF(RADIC.LT.0.0)GO TO 75 TRD00950
PNEAR=CLOSE-SQRT(RADIC) TRD00960
PFAR=CLOSE+SQRT(RADIC) TRD00970
PLENSP=AMIN1(PFAR,XNORM)-AMAX1(PNEAR,0.0) TRD00980
IF(PLENSP.LT.0.0)PLENSP=0.0 TRD00990
ACLS=PLENSP*SPHAL/VOLSPH TRD01000
GO TO 76 TRD01010
75 ACLS=0.0 TRD01020
76 ACLSPH=ACLSPH+ACLS TRD01030
CALL TRNCHK(CWINDS,ACLWAK,ACLSPH) TRD01040
IF(TEST)GO TO 999 TRD01050
GO TO 80 TRD01060
C COMPUTE CONTRIBUTIONS FOR SPHERE AND WAKE FOR A HORIZONTAL PATH TRD01070
C TRD01080
110 IF(TR(3).GT.ZCM+RSPH)GO TO 999 TRD01090
XCEN=XCM+DIFF(1,J) TRD01100
YCEN=YCM+DIFF(2,J) TRD01110
IF(TR(3).LT.ZCM-RSPH)GO TO 130 TRD01120
RADIUS=SQRT(RSPH**2-(TR(3)-ZCM)**2) TRD01130
CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENSP) TRD01140
IF(PLENSP.LT.0.0)PLENSP=0.0 TRD01150
ACLSPH=ACLSPH+PLENSP*SPHAL/VOLSPH TRD01160
CALL TRNCHK(CWINDS,ACLWAK,ACLSPH) TRD01170
IF(TEST)GO TO 999 TRD01180
130 IF(TR(3).GT.ZCM)GO TO 999 TRD01190
IF(TR(3).LE.5.0)GO TO 999 TRD01200
ZETA=(TR(3)-5.0)/(ZCM-5.0) TRD01210
XCEN=ZETA*XCEN+(1-ZETA)*XB TRD01220
YCEN=ZETA*YCEN+(1-ZETA)*YB TRD01230
RADIUS=ZETA*RSPH+(1-ZETA)*RB TRD01240
CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENWK) TRD01250
IF(PLENWK.LT.0.0)PLENWK=0.0 TRD01260
ACLWAK=ACLWAK+PLENWK*WAKAL/VOLWAK TRD01270
CALL TRNCHK(CWINDS,ACLWAK,ACLSPH) TRD01280
IF(TEST)GO TO 999 TRD01290
80 CONTINUE TRD01300
999 RETURN TRD01310
END TRD01320

```

```
SUBROUTINE UNIT(A,B,XNORM)          UNIT0010
  DIMENSION A(2),B(2)                UNIT0020
C *** B IS THE NORM OF A, AND XNORM IS THE MAGNITUDE
  XNORM=SQRT(A(1)**2+A(2)**2)        UNIT0030
  B(1)=A(1)/XNORM                  UNIT0040
  B(2)=A(2)/XNORM                  UNIT0050
  RETURN                            UNIT0060
END                                UNIT0070
                                    UNIT0080
```

```

SUBROUTINE VEHCL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THUND,PHI,NSOIL,      VCL00160
1           SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR,      VCL00170
2           VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH,      VCL00180
3           TRNLOS,NERR)      VCL00190
VCL00010
VCL00020
VCL00030
VCL00040
VCL00050
VCL00060
VCL00070
VCL00080
VCL00090
VCL00100
VCL00110
VCL00120
VCL00130
VCL00140
VCL00150
VCL00200
VCL00210
VCL00220
VCL00230
VCL00240
VCL00250
VCL00260
VCL00270
VCL00280
VCL00290
VCL00300
VCL00310
VCL00320
VCL00330
VCL00340
VCL00350
VCL00360
VCL00370
VCL00380
VCL00390
VCL00400
VCL00410
VCL00420
VCL00430
VCL00440
VCL00450
VCL00460
VCL00470
VCL00480
VCL00490
VCL00500
VCL00510
VCL00520
VCL00530
VCL00540
VCL00550
VCL00560
VCL00570
VCL00580
VCL00590
VCL00600
VCL00610
VCL00620
VCL00630
VCL00640
VCL00650
VCL00660
VCL00670
VCL00680
VCL00690
VCL00700
C THIS ROUTINE CONTROLS THE FLOW OF THE CALCULATION FOR THE
C VEHICLE GENERATED DUST CLOUD.
C INPUTS
C   SEE DRTRAN FOR DETAILS
C OUTPUTS
C   TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT
C   NERR   - ERROR CODE
*****
DIMENSION ZTMP(2), TMPMES(2), ZWND(2), WNDMES(2)      VCL00200
DIMENSION TRNCOR(3), RECCOR(3), TRN(3), REC(3), TRNFRM(2,2), ORIG(2),      VCL00210
1           V1(2), V0(2), VDIR(2)      VCL00220
LOGICAL DHDT,ERR,NEWATM,NEWVEH      VCL00230
INTEGER VEHTYP      VCL00240
COMMON /IOUNIT/IOUT,IIN,IPHUN,LOUNIT,NBIRTU,NCLIMT,KSTOR,NPLOTUVCL00250
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTP1,SCRNC2)      VCL00260
COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINY      VCL00270
COMMON/CONST/P1,PI2,PIRAD,TWOP1,TORRMB,CDEGK      VCL00280
DATA RTP1 /1.772453/
THETAX=THUND*PIRAD      VCL00290
IFC(.NOT.NEWATM)GO TO 100      VCL00300
CALL ATMCAL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR)      VCL00310
IFC(.NOT.ERR)GO TO 100      VCL00320
NERR=?
GO TO 999
100 CONTINUE
IFC(.NOT.NEWVEH)GO TO 5      VCL00330
CALL VSRC(VEHSPD,VEHWID,VEHWHT,VEHTYP,NSOIL,SILT)      VCL00340
CALL PREVEH(NSOIL,NWL)      VCL00350
5 CONTINUE
C COMPUTE DIRECTION VECTOR FOR THE VEHICLE MOTION FROM USERS INPUT      VCL00400
C
ANGL=VEHDIR*PIRAD      VCL00410
V1(1)=VEHSPD*COS(ANGL)      VCL00420
V1(2)=VEHSPD*SIN(ANGL)      VCL00430
C COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER      VCL00440
C DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN      VCL00450
C THE WIND DIRECTION.      VCL00460
C
TRNFRM(1,1)=COS(THETAX)      VCL00470
TRNFRM(2,2)=TRNFRM(1,1)      VCL00480
TRNFRM(1,2)=SIN(THETAX)      VCL00490
TRNFRM(2,1)=-TRNFRM(1,2)      VCL00500
ORIG(1)=V0(1)      VCL00510
ORIG(2)=V0(2)      VCL00520
C COMPUTE NEW COORDINATES BY MULTIPLYING BY THE TRANSFORMATION MATRIX      VCL00530
C
TRN(3)=TRNCOR(3)
REC(3)=RECCOR(3)
DO 20 I=1,2
TRN(I)=0.0
REC(I)=0.0
VDIR(I)=0.0
DO 10 J=1,2
TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))
REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))
VDIR(I)=VDIR(I)+TRNFRM(I,J)*V1(J)
20 CONTINUE
10 CONTINUE

```

10 CONTINUE
20 CONTINUE
C CALL VEHTRN ROUTINE TO USE COMPUTED QUADRATIC FITS TO CALCULATE
C A TRANSMITTANCE
C CALL VEHTRN(TRN,REC,TIME,VDIR,TRNL0S)
999 RETURN
END

VCL00710
VCL00720
VCL00730
VCL00740
VCL00750
VCL00760
VCL00770
VCL00780
VCL00790

```

SUBROUTINE VEHTRNK(TRN,REC,TIME,VDIR,TRNLOS) VTN00340
C THIS ROUTINE PARAMETERIZES THE LINE CONNECTING THE TRANSMITTER VTN00010
C AND RECEIVER IN THE LOCAL COORDINATE SYSTEM AND DOES A TRAPEZOIDAL VTN00020
C INTEGRATION FROM VEHICLE TIME=0.0 TO VEHICLE TIME=TIME THE VTN00030
C TRANSMITTANCE IS DESIRED. VTN00040
VTN00050
VTN00060
VTN00070
VTN00080
VTN00090
VTN00100
VTN00110
VTN00120
VTN00130
VTN00140
VTN00150
VTN00160
VTN00170
VTN00180
VTN00190
VTN00200
VTN00210
VTN00220
VTN00230
VTN00240
VTN00250
VTN00260
VTN00270
VTN00280
VTN00290
VTN00300
VTN00310
VTN00320
VTN00330
VTN00340
VTN00350
VTN00360
VTN00370
VTN00380
VTN00390
VTN00400
VTN00410
VTN00420
VTN00430
VTN00440
VTN00450
VTN00460
VTN00470
VTN00480
VTN00490
VTN00500
VTN00510
VTN00520
VTN00530
VTN00540
VTN00550
VTN00560
VTN00570
VTN00580
VTN00590
VTN00600
VTN00610
VTN00620
VTN00630
VTN00640
VTN00650
VTN00660
VTN00670
VTN00680
VTN00690
VTN00700

```

INPUTS

- TRN - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE SYSTEM VTN00090
- REC - THE COORDINATES OF THE RECEIVER IN THE LOCAL COORDINATE SYSTEM VTN00120
- TIME - THE PRESENT TIME AT WHICH A TRANSMITTANCE IS DESIRED VTN00150
- NWL - INTEGER INDEX FOR WAVELENGTH BEING USED VTN00160
- NSOIL - INTEGER INDEX FOR SOIL TYPE VTN00180
- VDIR - VECTOR INDICATING THE DIRECTION AND SPEED OF THE VEHICLE IN THE LOCAL COORDINATE SYSTEM VTN00210

OUTPUT

- TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT AT THE INDICATED TIME VTN00250

FUNCTIONS AND SUBROUTINES NEEDED

- GRAND EVALUATES THE INTEGRAND VTN00310

```

*****LOGICAL TEST***** VTN00320
DIMENSION TRN(3),REC(3),TR(3),RE(3),VDIR(2),DWF(5,2),U(3) VTN00350
COMMON/M05/DMMY(604),DMMC(600),
+ ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25), VTN00370
+ RB(3,25),Z2(3,25)
COMMON/TRANNY/THRESH,TEST,NWL,NSOIL VTN00390
DATA DWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,4.E-03/
TEST=.FALSE. VTN00410
VTN00420

```

PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER

```

XNORM=0.0 VTN00430
DO 10 I=1,3 VTN00440
RE(I)=REC(I) VTN00450
TR(I)=TRN(I) VTN00460
U(I)=RE(I)-TR(I) VTN00470
XNORM=XNORM+U(I)**2 VTN00480
10 CONTINUE VTN00490
XNORM=SQRT(XNORM) VTN00500
U(1)=U(1)/XNORM VTN00510
U(2)=U(2)/XNORM VTN00520
U(3)=U(3)/XNORM VTN00530

```

INCREMENT VEHICLE TRAVEL TIME AND CALL GRAND TO COMPUTE THE VALUE OF THE INTEGRAND. IF THE VALUE OF THE VARIABLE TIME IS GREATER THAN THE MAXIMUM TIME THAT HAS BEEN STORED (APPROX. 373 SEC) THEN ANY DUST PRODUCED MORE THAN TIMES(20) SECONDS IS ASSUMED TO HAVE NO EFFECT ON THE TRANSMITTANCE.

```

IF(TIME.LE.TIMES(ICOUNT))GO TO 11 VTN00580
TINC=TIMES(ICOUNT)/400. VTN00590
TSTART=TIME-TIMES(ICOUNT) VTN00600
GO TO 15 VTN00610
11 TINC=TIME/400. VTN00620
TSTART=0.0 VTN00630

```

```

15 SUM=0.0
DO 50 I=1,401
TIVEH=TSTART+FLOAT(I-1)*TINC
CALL GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)
IF(I.EQ.1.OR.I.EQ.401)GO TO 20
SUM=SUM+VALUE
GO TO 40
20 SUM=SUM+VALUE/2.
40 CONTINUE
SUM1=SUM*TINC
ACLWAK=0.0
ACLSPH=0.0
CALL TRNCHK(SUM1,ACLWAK,ACLSPH)
IF(TEST)GO TO 998
50 CONTINUE
SUM=SUM*TINC
ACL=UWF(NWL,NSOIL)*SUM
TRNLDS=EXP(-ACL)
GO TO 999
998 TRNLDS=0.0
999 RETURN
END

```

VTN00710
VTN00720
VTN00730
VTN00740
VTN00750
VTN00760
VTN00770
VTN00780
VTN00790
VTN00800
VTN00810
VTN00820
VTN00830
VTN00840
VTN00850
VTN00860
VTN00870
VTN00880
VTN00890
VTN00900
VTN00910
VTN00920

```

SUBROUTINE VSRC(VSPD,VWID,VWHT,VEHTYP,NSOIL,SILT) VRS00320
THIS SUBROUTINE INITIALIZES THE DUST CLOUD PRODUCED BY A VEHICLE VRS00010
INPUTS VRS00020
      VRS00030
      VRS00040
      VRS00050
      VRS00060
      VRS00070
      VRS00080
      VRS00090
      VRS00100
      VRS00110
      VRS00120
      VRS00130
      VRS00140
      VRS00150
      VRS00160
      VRS00170
      VRS00180
      VRS00190
      VRS00200
      VRS00210
      VRS00220
      VRS00230
      VRS00240
      VRS00250
      VRS00260
      VRS00270
      VRS00280
      VRS00290
      VRS00300
***** VRS00310
INTEGER VEHYP VRS00330
COMMON /PRINF/R0,VGRAV(3),NPRTS VRS00340
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUVRS00350
COMMON /DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)VRS00360
COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV VRS00370
COMMON /EKTEMP/Z0,ZL,T0,TC1,TC2,TC3 VRS00380
COMMON /PRE/Z,RT2DZ VRS00390
COMMON /YL/YLOAD VRS00400
NDSCS=1 VRS00410
NPRTS=1 VRS00420
VGRAV(1)=0.0 VRS00430
QDSC(1,1)=1.0 VRS00440
VRS00450
C INITIALIZE THE VEHICLE SOURCE FOR WHEELED VEHICLES VRS00460
      ZDSC(1)=VWID/8. VRS00470
      Z=VWID/4. VRS00480
      ZZ=VWID/4. VRS00490
      DZ=DIFFUS(Z0,ZL,ZZ) VRS00500
      DX=DXZ0*DZ VRS00510
      TDSC(1)=-3.*((VWID**2)/512./DZ) VRS00520
      TOF=-TDSC(1) VRS00530
      CALL MOMENT(VGRAV,Z,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2) VRS00540
      XDSC(1)=-XBAR VRS00550
      A=1.0 VRS00560
      B=SIGW2+SIGP2 VRS00570
      C=4.*((SIGW2+SIGP2)-((VWID/3.)**4)) VRS00580
      RAD=B**2-A*C VRS00590
      R2DSC(1)=-B+SQRT(RAD) VRS00600
      RT2DZ=SQRT(2.*DZ) VRS00610
VRS00620
C INITIALIZE LOADING FOR VEHICLE (YLOAD IN KG/SEC) VRS00630
      20 SILTPC=100.*SILT VRS00640
      A=3.8E-9 VRS00650
      IF(VEHTYP.GT.0)A=1.52E-08 VRS00660
      Q=A*VSPD*VWHT*SILTPC VRS00670
      ALPHA=240. VRS00680
      VRS00690
      VRS00700

```

VLOAD=ALPHA*VSPD*Q
999 RETURN
END

VRS00710
VRS00720
VRS00730

```
SUBROUTINE VSUM(A,B,S,C)          VSU00010
DIMENSION A(2),B(2),C(2)          VSU00020
C *** C=A+S*B WHERE A,B,C ARE VECTORS AND S IS SCALAR   VSU00030
DO 14 J=1,2                      VSU00040
14 C(J)=A(J)+S*B(J)              VSU00050
RETURN                           VSU00060
END                             VSU00070
```

```

SUBROUTINE WIN(Z,U,V)                               WIN00100
COMMON/STARS/USTAR,TSTAR,ZSTAR                   WIN00200
COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT           WIN00300
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3             WIN00400
COMMON/BUOYCL/Y(8),SPHNS(3),RISTIM            WIN00500
*****                                              WIN00600
WIN00700
WIN00800
WIN00900
WIN01000
WIN01100
WIN01200
WIN01300
WIN01400
WIN01500
WIN01600
WIN01700
WIN01800
WIN01900
WIN02000
WIN02100
WIN02200
WIN02300
WIN02400
WIN02500
WIN02600
WIN02700
WIN02800
WIN02900
WIN03000
WIN03100
WIN03200
WIN03300
WIN03400
WIN03500
WIN03600
WIN03700
WIN03800

```

PURPOSE

TO COMPUTE THE WIND SPEED AT A SPECIFIED HEIGHT

INPUTS

Z HEIGHT AT WHICH WIND SPEEDS ARE DESIRED

OUTPUTS

U WIND SPEED IN THE DIRECTION OF THE GROUND WIND

V WIND SPEED PERPENDICULAR TO THE GROUND WIND

CALLED BY DIFEQ

SUBROUTINES AND FUNCTIONS NEEDED

WNDCAL CALCULATES SCALED WIND SPEED

```

*****                                              WIN02800
IF(Z.GT.ZSTAR)GO TO 100
U=USTAR*WNDCAL(Z0,ZL,Z)
V=0.0
GO TO 999
100 UE=C*EXP(-ALP*Z)*COS(ALP*Z)-PYF
VE=-C*EXP(-ALP*Z)*SIN(ALP*Z)+PXF
U=UHAT*UE+VHAT*VE
V=-VHAT*UE+UHAT*VE
999 RETURN
END

```

```

FUNCTION WNDCAL(Z0,ZL,Z)
***** WND00010
      WND00020
      WND00030
      WND00040
      WND00050
      WND00060
      WND00070
      WND00080
      WND00090
      WND00100
      WND00110
      WND00120
      WND00130
      WND00140
      WND00150
      WND00160
      WND00170
      WND00180
      WND00190
      WND00200
      WND00210
      WND00220
      WND00230
      WND00240
      WND00250
      WND00260
      WND00270
      WND00280
      WND00290
      WND00300
      WND00310
      WND00320
      WND00330
      WND00340
      WND00350
      WND00360
      WND00370
      WND00380
      WND00390
      WND00400
      WND00410
      WND00420
      WND00430
      WND00440
      WND00450
      WND00460
      WND00470
      WND00480
      WND00490
      WND00500
      WND00510
      WND00520
      WND00530
      WND00540
      WND00550
      WND00560
      WND00570
      WND00580
      WND00590
      WND00600
      WND00610
      WND00620
      WND00630
      WND00640
      WND00650
      WND00660
      WND00670
      WND00680
      WND00690
      WND00700

PURPOSE
TO CALCULATE THE WIND SPEED, U/U*, SCALED BY THE FRICTION
VELOCITY FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV LENGTH AT A
SPECIFIED HEIGHT.
INPUTS
Z0      THE FRICTION HEIGHT IN METERS.
ZL      THE MONIN-OBUKHOV LENGTH IN METERS.
Z      THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
IN METERS

RETURNS VELOCITY SCALED BY FRICTION VELOCITY
CALLED BY ATMCAL,WIN AND RISE
***** WND00260
LOGICAL LOW
COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
PSIM(Z,21)= ALOG((1.-Z)/(1.-Z1))-ALOG((1.+Z)/(1.+Z1))+$2.*ATAN(Z)-ATAN(Z1))
PSIMS(Z)=-7.*Z
PHIM(Z)=(1.-16.*Z)**(-.25)

PHIM      THE SHEAR OF MOMENTUM
PSIM      THE UNIVERSAL FUNCTION FOR THE DEVIATION FROM
          LOGARITHMIC WIND VELOCITY BOUNDARY LAYER PROFILE IN AN
          UNSTABLE ATMOSPHERE
PSIMS    THE SAME AS PSIM FOR A STABLE ATMOSPHERE

IF(ABS(ZL),LE.1,E3)GO TO 100
WNDCAL=ALOG(1.+Z/20)
GO TO 999
100 CONTINUE
P=SIGN(1.,ZL)
LOW=.TRUE.
S=Z/ZL
IF(S.LE.1.5,AND,S.GE.-2.)GO TO 10
S=AMIN1(S,1.5)
S=AMAX1(S,-2.)
LOW=.FALSE.
10 CONTINUE
IF(P>120,130,130
120 S=1./PHIM(S)
S1=Z0/ZL
S0=1./PHIM(S1)
WNDCAL=PSIM(S,S0)

DETERMINE THE CONSTANTS FOR MATCHING AT Z/ZL=-2.
S2=-2.
AW=-3.*((1-16.*(S2))**(-.25))*((-S2)**(1./3.))
CW=-1.*AW*(-S2)**(-1./3.)
GO TO 52
130 CONTINUE
PSI=PSIMS(S)
WNDCAL=ALOG(1.+S*ZL/Z0)-PSI

FIND THE CONSTANTS FOR MATCHING OF STABLE PROFILE AT Z/ZL=1.5

```

```
S2=1.5          WND00710
BW=1./(<Z0/ZL+S2>)+7.  WND00720
DW=-1.*S2*BW  WND00730
52 CONTINUE    WND00740
IF(LOW>GO TO 999  WND00750
IFCP>53,53,54  WND00760
53 WNDCAL=WNDCAL+CW+AW*(-ZL/Z)**(1./3.)
54 WNDCAL=WNDCAL+DW+BW*Z/ZL  WND00770
999 WNDCAL=WNDCAL/.4  WND00780
998 CONTINUE    WND00790
RETURN        WND00800
END           WND00810
                  WND00820
                  WND00830
```

```

SUBROUTINE NMMW(FREQGH, ICLMAT, MMTRAN, IERR) NMM00010
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK NMM00020
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT, NMM00030
+ FOGPRE,WNDVEL,WNDDIR,IPASCT NMM00040
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUNMM00050
COMMON /GEOMET/PTS(15),IGEOSW NMM00060
C----- NMM00070
C PURPOSE: TO CALCULATE THE EXTINCTION AND BACKSCATTER AT MILLIMETER NMM00080
FREQUENCIES 10 TO 1000 GHZ, DUE TO NMM00090
GASEOUS ABSORPTION, FOG(ICE AND WATER) AND NMM00100
CLOUD BULK ATTENUATION, RAIN, SNOW EXTINCTION. NMM00110
C----- NMM00120
C** INPUT TO THE NMMW MODULE IS PERFORMED THROUGH A CARD ORDER- NMM00130
C** INDEPENDENT INPUT TECHNIQUE. A FOUR-LETTER IDENTIFIER IN COLS. NMM00140
C** 1-4 OF EACH RECORD SPECIFIES THE TYPE OF DATA BEING READ BY THE NMM00150
C** MODULE. THE INPUT CARDS MAY APPEAR IN ANY ORDER WITH THE EXCEPTION NMM00160
C** OF OF THE (GO) END OF READ SENTINEL, WHICH MUST BE THE LAST CARD NMM00170
C** READ. ALL OF THE FOLLOWING CARDS ARE READ IN UNDER THE FORMAT NMM00180
C** (A4,1X,3(E10.4,1X)) : NMM00190
NMM00200
NMM00210
NMM00220
NMM00230
NMM00240
NMM00250
NMM00260
NMM00270
NMM00280
NMM00290
NMM00300
NMM00310
NMM00320
NMM00330
NMM00340
NMM00350
NMM00360
NMM00370
NMM00380
NMM00390
NMM00400
NMM00410
NMM00420
NMM00430
NMM00440
NMM00450
NMM00460
NMM00470
NMM00480
NMM00490
NMM00500
NMM00510
NMM00520
NMM00530
NMM00540
NMM00550
NMM00560
NMM00570
NMM00580
NMM00590
NMM00600
NMM00610
NMM00620
NMM00630
NMM00640
NMM00650
NMM00660
NMM00670
NMM00680
NMM00690
NMM00700
C----- NMM00710
C CARD IDENTIFIER : PATH NMM00720
C VARIABLES READ : MMWPTH <PATH LENGTH (KM)> NMM00730
C----- NMM00740
C CARD IDENTIFIER : ATMO NMM00750
C VARIABLES READ : TEMP1 <TEMPERATURE (DEG C)> NMM00760
C PRESS1 <PRESSURE (MB)> NMM00770
C ABSHUM <IF .GT. 0, ABSOLUTE NMM00780
C HUMIDITY (GM/M**3)> NMM00790
C <IF .LT. 0, RELATIVE NMM00800
C HUMIDITY (%)> NMM00810
C----- NMM00820
C ***NOTE: TEMP1, PRESS1, ABSHUM WILL BE PASSED FROM CLIMAT IF NMM00830
ICLMAT=1. IN THAT EVENT, THE (ATMO) CARD IS NOT NEEDED. NMM00840
C----- NMM00850
C CARD IDENTIFIER : FOGD NMM00860
C VARIABLES READ : FOGDEN <FOG DENSITY (LIQUID WATER, NMM00870
GM/M**3)> NMM00880
NMM00890
NMM00900
NMM00910
NMM00920
NMM00930
NMM00940
NMM00950
NMM00960
NMM00970
NMM00980
NMM00990
NMM01000
NMM01010
NMM01020
NMM01030
NMM01040
NMM01050
NMM01060
NMM01070
NMM01080
NMM01090
NMM01100
NMM01110
NMM01120
NMM01130
NMM01140
NMM01150
NMM01160
NMM01170
NMM01180
NMM01190
NMM01200
NMM01210
NMM01220
NMM01230
NMM01240
NMM01250
NMM01260
NMM01270
NMM01280
NMM01290
NMM01300
NMM01310
NMM01320
NMM01330
NMM01340
NMM01350
NMM01360
NMM01370
NMM01380
NMM01390
NMM01400
NMM01410
NMM01420
NMM01430
NMM01440
NMM01450
NMM01460
NMM01470
NMM01480
NMM01490
NMM01500
NMM01510
NMM01520
NMM01530
NMM01540
NMM01550
NMM01560
NMM01570
NMM01580
NMM01590
NMM01600
NMM01610
NMM01620
NMM01630
NMM01640
NMM01650
NMM01660
NMM01670
NMM01680
NMM01690
NMM01700
C----- NMM01710
C LOCAL VARIABLES NMM01720
C----- NMM01730
REAL MMWPTH,MMTRAN,MMBSXS
DIMENSION DAT(3),IAL(6)
DATA IAL/2HPA ,2HAT ,2HFO ,2HRA ,2HSN ,2HGO /
IERR=0
GASABS=0.
FOGEXT=0.
RAINEX=0.
FOGBS=0.
RAINSB=0.
SNOWBS=0.
5 READ(I0IN,400)IALFA,IALFA2,(DAT(L),L=1,3)
400 FORMAT(2A2,1X,3(E10.4,1X))
IF(IALFA.EQ.IAL(1)) GO TO 10
IF(IALFA.EQ.IAL(2)) GO TO 20

```

```

IF<IALFA.EQ.IAL<3>> GO TO 30 NMM00710
IF<IALFA.EQ.IAL<4>> GO TO 40 NMM00720
IF<IALFA.EQ.IAL<5>> GO TO 50 NMM00730
IF<IALFA.EQ.IAL<6>> GO TO 60 NMM00740
WRITE<I00UT,450>IALFA,IALFA2 NMM00750
450 FORMAT<1H0,20X,2A2,22H IS AN INCORRECT LABEL //>
GO TO 300 NMM00760
10 MMWPTH=DAT<1> NMM00770
GO TO 5 NMM00790
20 TEMP1=DAT<1> NMM00800
PRESS1=DAT<2> NMM00810
ABSHUM=DAT<3> NMM00820
GO TO 5 NMM00830
30 FOGDEN=DAT<1> NMM00840
GO TO 5 NMM00850
40 RAINRT=DAT<1> NMM00860
GO TO 5 NMM00870
50 SNOWRT=DAT<1> NMM00880
GO TO 5 NMM00890
60 CONTINUE NMM00900
IF<IGEUSW.NE.1>GO TO 99 NMM00910
MMWPTH=SQRT<(PTS<4>-PTS<1>)**2+(PTS<5>-PTS<2>)**2+
+(PTS<6>-PTS<3>)**2> NMM00920
99 CONTINUE NMM00930
IF<ICLMAT.EQ.1> TEMP1=TEMP NMM00940
IF<ICLMAT.EQ.1> PRESS1=PRESS NMM00950
IF<ICLMAT.EQ.1> ABSHUM=AH NMM00960
WRITE<I00UT,500> TEMP1,PRESS1,ABSHUM,FOGDEN NMM00980
WRITE<I00UT,600> RAINRT,SNOWRT,FREQGH,MMWPTH NMM00990
C CHECK MODEL INPUTS FOR RANGE OF VALIDITY. NMM01000
IF<PRESS1.GE.500.> GO TO 100 NMM01010
IERR=1 NMM01020
WRITE<I00UT,800> NMM01030
100 IF<FREQGH.LE.1000..AND.FREQGH.GE.10.> GO TO 150 NMM01040
IERR=1 NMM01050
WRITE<I00UT,900> NMM01060
150 IF<IERR.EQ.1> GO TO 300 NMM01070
C CHANGE UNITS NMM01080
PRSTOR=PRESS1/TORRMB NMM01090
TEMPDK=TEMP1+CDEGK NMM01100
C CALL INDIVIDUAL MODULES FOR GAS, FOG/CLOUD, SNOW, RAIN EXTINCTION NMM01110
CALL MMWGS<TEMPDK,PRSTOR,ABSHUM,FREQGH,GASABS> NMM01120
IF<FOGDEN.GT.1.E-10> NMM01130
+ CALL MMWFG<FOGDEN,TEMPDK,FREQGH,FOGEXT,FOGBS> NMM01140
+ IF<RAINRT.GT.1.E-10> NMM01150
+ CALL MMRAN<RAINRT,TEMPDK,FREQGH,2.,RAINEX,RAINBS> NMM01160
IF<SNOWRT.GT.1.E-10> NMM01170
+ CALL MMSNO<SNOWRT,TEMPDK,FREQGH,SNOWEX,SNOWBS> NMM01180
C COMPUTE TRANSMISSION NMM01190
TOTEXT=GASABS+FOGEXT+RAINEX+SNOWEX NMM01200
MMBSXS=FOGBS+RAINBS+SNOWBS NMM01210
MMTRAN=EXP<(-MMWPTH*TOTEXT)> NMM01220
C CHANGE UNITS FOR ABSORPTION/EXTINCTION FROM 1/KM TO DB/KM. NMM01230
DBKM=4.343 NMM01240
GASABS=GASABS*DBKM NMM01250
FOGEXT=FOGEXT*DBKM NMM01260
RAINEX=RAINEX*DBKM NMM01270
SNOWEX=SNOWEX*DBKM NMM01280
WRITE<I00UT,700> GASABS,FOGEXT,RAINEX,SNOWEX,MMTRAN NMM01290
WRITE<I00UT,750> FOGBS,RAINBS,SNOWBS,MMBSXS NMM01300
C COMPUTATION COMPLETED NMM01310
300 RETURN NMM01320
NMM01330
500 FORMAT<1H0,///,47X,12HTEMPERATURE,14X,F8.3, NMM01340
+ 10H DEGREES C,/,47X,9HPRESSURE,17X,F8.3, NMM01350
+ 3H MB,/,47X,17HABSOLUTE HUMIDITY,9X,F8.3, NMM01360
+ 7H G/M**3/,47X,11HFOG DENSITY,15X,F8.3, NMM01370
+ 7H G/M**3> NMM01380
600 FORMAT<1H,46X,9HRAIN RATE,17X,F8.3,6H MM/HR,/ NMM01390
+ ,47X,9HSNOW RATE,17X,F8.3,6H MM/HR,/ NMM01400

```

```

+      ,47X,9HFREQUENCY,17X,F8.3,4H GHZ/
+      ,47X,11H PATH LENGTH,15X,F8.3,3H KM)
700  FORMAT(1H0,46X,14HGAS ABSORPTION,10X,E10.4,6H DB/KM,/
+      ,47X,15H FOG EXTINCTION ,9X,E10.4,6H DB/KM,/
+      ,47X,16H RAIN EXTINCTION ,8X,E10.4,6H DB/KM,/
+      ,47X,16H SNOW EXTINCTION ,8X,E10.4,6H DB/KM,/
+      ,47X,13H TRANSMISSION ,11X,E10.4/)
750  FORMAT(1H ,46X,15H FOG BACKSCATTER,9X,E10.4,10H M**2/M**3,/
+      ,47X,16H RAIN BACKSCATTER,8X,E10.4,10H M**2/M**3,/
+      ,47X,16H SNOW BACKSCATTER,8X,E10.4,10H M**2/M**3,/
+      ,47X,17H TOTAL BACKSCATTER,7X,E10.4,10H M**2/M**3)
800  FORMAT(1H0,47X,41H PRESSURE LESS THAN 500 MB, GAS ABSORPTION,
+      ,19H WILL BE INACCURATE)
900  FORMAT(1H0,47X,31H FREQUENCY<10, GHZ, OR >1000 GHZ,
+      ,42H CALCULATION WILL FAIL, USE OTHER FREQUENCY)
      END

```

NMM01410
NMM01420
NMM01430
NMM01440
NMM01450
NMM01460
NMM01470
NMM01480
NMM01490
NMM01500
NMM01510
NMM01520
NMM01530
NMM01540
NMM01550
NMM01560

```
FUNCTION AB(WA,A,CE,B,C)
AB=A*EXP(-ABS((ALOG10(1.E4*WA/CE)/B))**C)
RETURN
END
```

```
AB 00010
AB 00020
AB 00030
AB 00040
```

```
FUNCTION DOP(WA,A,CE1,B,C,CE2,D,E,CE3,F,G)
V=1./WA
V2=V*V
H1=CE1*CE1-V2
H2=CE2*CE2-V2
H3=CE3*CE3-V2
DOP=SQRT(A+B*H1/(H1+H1+C*V2)+D*H2/(H2+H2+E*V2)+F*H3/(H3+H3+G*V2))
RETURN
END
```

DOP000010
DOP000020
DOP000030
DOP000040
DOP000050
DOP000060
DOP000070
DOP000080
DOP000090

```

SUBROUTINE INTRP(A,B,T,F,TT,FF,AA,BB,J)           INT00010
C-----INT00020
C-----INT00030
C-----PURPOSE: TO DO FREQUENCY AND TEMPERATURE INTERPOLATION   INT00040
C-----INT00050
C-----DIMENSION A(9,3), B(9,3), F(9), T(3)           INT00060
C-----INT00070
C-----INT00080
C-----IF(TT.LT.T(1)) TT=T(1)           INT00090
C-----DO 11 J=2,3           INT00100
C----- IF(TT.LT.T(J)) GO TO 14           INT00110
11 CONTINUE           INT00120
TT=T(3)           INT00130
J=3           INT00140
14 CONTINUE           INT00150
DO 15 I=2,9           INT00160
IF(FF.LT.F(I)) GO TO 16           INT00170
15 CONTINUE           INT00180
FF=F(9)           INT00190
I=9           INT00200
16 FF=ALOG10(FF)           INT00210
F0=ALOG10(F(I))           INT00220
F1=ALOG10(F(I-1))           INT00230
FF0=(F0-FF)/(F0-F1)           INT00240
FF1=(FF-F1)/(F0-F1)           INT00250
TF0=(T(J)-TT)/(T(J)-T(J-1))           INT00260
TF1=(TT-T(J-1))/(T(J)-T(J-1))           INT00270
A11=ALOG10(A(I-1,J-1))           INT00280
A01=ALOG10(A(I,J-1))           INT00290
A10=ALOG10(A(I-1,J))           INT00300
A00=ALOG10(A(I,J))           INT00310
APJ1=A11*FF0+A01*FF1           INT00320
APJ0=A10*FF0+A00*FF1           INT00330
AA=APJ1*TF0+APJ0*TF1           INT00340
B11=ALOG10(B(I-1,J-1))           INT00350
B10=ALOG10(B(I,J-1))           INT00360
B01=ALOG10(B(I-1,J))           INT00370
B00=ALOG10(B(I,J))           INT00380
BPJ1=B11*FF0+B01*FF1           INT00390
BPJ0=B10*FF0+B00*FF1           INT00400
BB=BPJ1*TF0+BPJ0*TF1           INT00410
AA=10.**AA           INT00420
BB=10.**BB           INT00430
RETURN           INT00440
END

```

```

SUBROUTINE MMH20(V,T,PTOT,PH20,DATH20,ABH20)           MMMH00010
*****                                                       MMMH00020
ROUTINE TO CALCULATE H20 VAPOR ABSORPTION FOR 0 TO 350 GHZ.   MMMH00030
*****                                                       MMMH00040
INPUTS ARE: WAVENUMBER(/CM), TEMPERATURE(KELVIN), TOTAL    MMMH00050
PRESSURE(TORR), H20 VAPOR PRESSURE(TORR), LINE DATA ARRAY.   MMMH00060
OUTPUT IS: H20 VAPOR ABSORPTION                           MMMH00070
*****                                                       MMMH00080
CALLED BY MMWGS   MAKES NO CALLS                         MMMH00090
*****                                                       MMMH00100
LOCAL VARIABLES:                                         MMMH00110
WCD   VAPOR COLUMN DENSITY(/CM/CM/KM)                   MMMH00120
CT    LINE STRENGTH TEMPERATURE CORRECTION             MMMH00130
CA    LINE WIDTH SELF BROADENING AND TEMP. CORRECTION   MMMH00140
SA    CORRECTED LINE STRENGTH                          MMMH00150
GA    CORRECTED LINE WIDTH                            MMMH00160
ABS   SINGLE LINE ABSORPTION(/KM)                      MMMH00170
*****                                                       MMMH00180
DIMENSION DATH20(37,4)                                 MMMH00190
*****                                                       MMMH00200
C
      ABH20=0.                                         MMMH00210
      WCD=7.33994E26*PH20/760./T*(PFR(T))            MMMH00220
      CT=4.860773E-3*(T-296.)/T                      MMMH00230
      CA=((296./T)**.62)*(PTOT+(4.*PH20))/760.        MMMH00240
C
      DO 500 L=1,37
      SA=DATH20(L,2)*WCD*EXP(DATH20(L,4)*CT)          MMMH00250
      GA=DATH20(L,3)*CA                                MMMH00260
      ABS=SA*SUPK(V,DATH20(L,1),GA)                  MMMH00270
      ABH20=ABH20+ABS
500  CONTINUE
      RETURN
      END

```

```

C   SUBROUTINE MMIDX(XL,T,ICE,H20AB,H20K2)           MMI00010
C   REF: RAY, APPLIED OPTICS, VOL. 11, P. 1836,(1972)   MMI00020
      COMPLEX CINDEX,XK                                MMI00030
      XXL=XL/10.                                         MMI00040
      TT=T-273.16                                       MMI00050
      T3=TT-25.                                         MMI00060
      IF(ICE.NE.0) GO TO 150                            MMI00070
C   PARAMETERS FOR WATER                           MMI00080
100   EFIN=5.27137+.0216474*TT-1.31198E-3*TT*TT    MMI00090
      ALFA=-16.8129/T+.0609265                         MMI00100
      XLS=3.3836E-4*EXP(-2513.98/T)                   MMI00110
      SIGMA=1.25664E9                                    MMI00120
      ES=78.54*(1.-4.579E-3*T3+1.19E-5*T3*T3-2.8E-8*T3*T3*T3)
      GO TO 200                                         MMI00130
C   PARAMETERS FOR ICE                           MMI00140
C   150   EFIN=3.168                                 MMI00150
      ALFA=0.288+0.0052*TT+2.3E-4*TT*TT               MMI00160
      XLS=9.990288E-5*EXP(1.32E4/(1.9869*T))        MMI00170
      SIGMA=1.26*EXP(-1.25E4/(1.9869*T))            MMI00180
      ES=203.168+2.5*TT+0.15*TT*TT                  MMI00190
200   U=(ES-EFIN)*(XLS/XXL)***(1.-ALFA)             MMI00200
      Y=1.+2.*((XLS/XXL)***(1.-ALFA))*SIN(ALFA*1.57079633)+MMI00210
      1. (XLS/XXL)***(2-2*ALFA)                         MMI00220
      EP=EFIN+((ES-EFIN)+U*SIN(ALFA*1.57079633))/Y   MMI00230
      EPP=(U+COS(ALFA*1.57079633))/Y+SIGMA*XXL/1.88496E11
      RE=SQRT((EP+SQRT(EP*EP+EPP*EPP))/2.)
      AI=-EPP/2./RE                                     MMI00240
      IF(ICE.NE.0) GO TO 400                            MMI00250
C   IF(XXL.LE..034) GO TO 307                      MMI00260
C   306   R2=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,MMI00270
      + 259913.,161.29,43319.7,27661.2)                MMI00280
      R2=R2+R2*T3*1.E-3*EXP((-2.5E-5*XXL)**.25)       MMI00290
      RE=RE*(XXL-.034)/.066+R2*(.1-XXL)/.066          MMI00300
      GO TO 311                                         MMI00310
307   RE=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,MMI00320
      + 259913.,161.29,43319.7,27661.2)                MMI00330
      RE=RE+RE*T3*1.E-3*EXP((-2.5E-5*XXL)**.25)       MMI00340
      CONTINUE                                           MMI00350
311   IF(XXL.GT..3) GO TO 500                      MMI00360
      AI=AI+AB(XXL,.25,300.,.47,3.)+AB(XXL,.39,17.,.45,1.3)MMI00370
      + AB(XXL,.41,62.,.35,1.7)                         MMI00380
      GO TO 500                                         MMI00390
C   400   CONTINUE                                           MMI00400
      IF(XXL.GT.0.08) GO TO 500                      MMI00410
405   RC=DOP(XXL,1.225,1652.9,1.12082E6,46E-11,909.09,416441.,119852.,MMI00420
      + 223.2,47031.8,126834.)                          MMI00430
      RE=RE*(XXL-0.02)/0.06+R2*(0.08-XXL)/0.06          MMI00440
      AI=AI+AB(XXL,.242,62.,.23,1.6)+AB(XXL,.581,44.8,0.055,1.)MMI00450
C   500   CINDEX=CMPLX(RE,AI)                        MMI00460
      XK=((CINDEX*CINDEX-1)/(CINDEX*CINDEX+2))         MMI00470
      H20AB=AIMAG(-XK)                                MMI00480
      H20K2=XK*CONJG(XK)                             MMI00490
      RETURN                                             MMI00500
      END                                               MMI00510
      MMI00520
      MMI00530
      MMI00540
      MMI00550
      MMI00560
      MMI00570
      MMI00580
      MMI00590

```

```

SUBROUTINE MMOXY(V,T,PTOT,PH20,DATA02,ABS02)          MM000010
C*****ROUTINE TO CALCULATE ABSORPTION DUE TO OXYGEN. METHOD IS THAT OF      MM000020
LIEBE, GIMMESTAD, & HOPPONEN. IEEE TRANS. ANT. PROP. V.25, P327.      MM000030
INPUTS ARE: FREQUENCY(GHZ), TEMPERATURE(KELVIN), TOTAL      MM000040
PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), O2 LINE DATA      MM000050
ARRAY.      MM000060
OUTPUTS ARE: O2 ABSORPTION (1/KM)      MM000070
CALLED FROM MMWGS CALLS NO OTHER ROUTINES.      MM000080
LOCAL VARIABLES:      MM000090
T2      300./T      MM000100
PHI      TEMPERATURE CORRECTION FOR LINE STRENGTHS      MM000110
S      CORRECTED LINE STRENGTH(HZ TORR)      MM000120
GAMMA      CORRECTED LINE WIDTH (1/GHZ.)      MM000130
XIF      LINE INTERFERENCE FACTOR      MM000140
VMI      DATA02(L,1)-V      MM000150
VPL      DATA02(L,1)+V      MM000160
PROFIL      MODIFIED VANVLECK-WEISSKOPF LINE SHAPE      MM000170
DIMENSION DATA02(42,6)      MM000180
C
ABS02=0,      MM000190
T2=300./T      MM000200
DO 500 L=1,42      MM000210
K=IFIX(DATA02(L,6))      MM000220
PHI=T2*T2*T2*EXP(-6.895E-3*K*(K+1)*(T2-1.))      MM000230
S=0.2045*PTOT*DATA02(L,2)*PHI      MM000240
GAMMA=DATA02(L,3)**(.929*PTOT*T2**.9+1.3*T2*PH20)*1.E-3      MM000250
XIF=DATA02(L,4)*T2**DATA02(L,5)*PTOT*1.E-3      MM000260
VMI=DATA02(L,1)-V      MM000270
VPL=VMI+2*V      MM000280
PROFIL=(V/DATA02(L,1))*((GAMMA-VMI*XIF)/(VMI+VMI+GAMMA+GAMMA))+      MM000290
*      (GAMMA-VPL*XIF)/(VPL*VPL+GAMMA*GAMMA))      MM000300
ABS02=ABS02+8*PROFIL*V*4.192E-5      MM000310
500 CONTINUE      MM000320
RETURN      MM000330
END      MM000340
MM000350
MM000360
MM000370
MM000380
MM000390
MM000400
MM000410
MM000420
MM000430
MM000440

```

```

SUBROUTINE MMRANK(RAINRT,T,FREQ,RTYPE,GRAIN,BSRAIN)
C REF: OLSEN, ET.AL., IEEE ANT. PROP., VOL.26, P.318(1978)          MMR00010
C ROUTINE TO COMPUTE THE ATTENUATION DUE TO RAIN, FOR           MMR00020
C FREQUENCIES BETWEEN 10 & 1000 GHZ; BACKSCATTER ALSO.           MMR00030
C INPUTS ARE: FREQUENCY(GHZ), TEMPERATURE(KELVIN), RAIN RATE(MM/HR) MMR00040
C OUTPUT IS ATTENUATION (1/KM), BACKSCATTER (MM**2/MM**3)        MMR00050
C CALLED FROM MMWMOD CALLS NO SUBROUTINES.                      MMR00060
C LOCAL VARIABLES:
C   ALFA,BETA          A AND B PARAMETERS FUNCTION OF FREQ,      MMR00070
C   BSAT,BSBT          TEMPERATURE, AND RAIN TYPE                 MMR00080
C                      POLYNOMIAL COEF.S FOR LINEAR FIT            MMR00090
C                      TO BACKSCATTER= ALF*RAINRT**BET             MMR00100
C----- DIMENSION ALFA(9,3,3), BETA(9,3,3), F(9), TK(3),A(9,3),B(9,3),      MMR00110
C + BSAT(6,2), BSBT(6,2)                                         MMR00120
C DATA ALFA/1.42E-2,3.34E-2,.197,.404,1.11,1.76,2.36,2.72,2.89,      MMR00130
C     1.14E-2,2.82E-2,.180,.387,1.18,1.89,2.46,2.78,2.98,      MMR00140
C     7.52E-3,2.20E-2,.167,.368,1.24,2.05,2.58,2.84,2.89,      MMR00150
C     1.59E-2,3.94E-2,.275,.579,1.35,1.80,2.07,2.22,2.00,      MMR00160
C     1.36E-2,3.68E-2,.268,.579,1.42,1.88,2.10,2.24,2.00,      MMR00170
C     1.01E-2,3.57E-2,.269,.572,1.46,1.96,2.13,2.24,2.01,      MMR00180
C     1.51E-2,4.20E-2,.376,.619,.871,1.05,1.02,972,.857,      MMR00190
C     1.69E-2,4.66E-2,.372,.629,.909,1.04,.997,.976,.856,      MMR00200
C     1.90E-2,5.89E-2,.360,.610,.936,1.02,.992,.997,.856/,      MMR00210
C----- BETA/.932,.954,1.016,1.027,.943,.846,.766,.704,.613,      MMR00220
C     .968,1.003,1.053,1.053,.941,.829,.749,.692,.612,      MMR00230
C     1.024,1.106,1.107,1.082,.936,.807,.725,.679,.613,      MMR00240
C     1.094,1.088,1.002,.904,.753,.686,.645,.618,.611,      MMR00250
C     1.150,1.118,1.007,.905,.742,.677,.641,.614,.610,      MMR00260
C     1.260,1.160,.999,.900,.732,.667,.635,.612,.610,      MMR00270
C     1.087,1.027,.784,.714,.657,.610,.608,.612,.614,      MMR00280
C     1.076,1.010,.783,.709,.650,.610,.613,.611,.615,      MMR00290
C     1.079,.966,.782,.708,.642,.615,.615,.607,.615/      MMR00300
C----- DATA TK/263.16,273.16,293.16/,      MMR00310
C + F/10,.15,.35,.50,.95,.140,.225,.310,.1000,/      MMR00320
C----- DATA BSAT/-,.8824881E+01,-,1029998E-01,+,.2451205E-04,      MMR00330
C + -.2462900E-07,+,.6507628E-10,-,1856080E-12,      MMR00340
C + -.2127020E+02,+,.6906017E+00,-,1924260E-01,      MMR00350
C + ,.3035233E-03,-,.2545323E-05,+,.8673581E-08/,      MMR00360
C----- BSBT/+,.7901887E+00,-,1900189E-02,+,.6341350E-05,      MMR00370
C + ,.3186429E-08,-,5933950E-10,+,.9056715E-13,      MMR00380
C + ,.1361993E+01,+,.3628100E-01,-,2461284E-02,      MMR00390
C + ,.4805257E-04,-,3988064E-06,+,.1193458E-08/      MMR00400
C----- TR=T
C----- GRAIN=0,
C----- BSRAIN=0,
C----- IF(FREQ.LT.10.) GO TO 200
C----- FR=FREQ
C----- ITYPE=IFIX(RTYPE+0.1)
C----- DO 10 I=1,3
C-----   DO 10 J=1,9
C-----     A(J,I)=ALFA(J,I,ITYPE)
C-----     B(J,I)=BETA(J,I,ITYPE)
C----- 10 CONTINUE
C----- CALL INTRP(A,B,TK,F,TR,FR,AA,BB,J)
C----- GRAIN=(AA*RAINRT**BB)/4.343
C----- CALCULATIONS FOR RAIN BACKSCATTER.
C----- AA=0.
C----- BB=0.
C----- IA=2
C----- IB=2

```

```
IF(FREQ.GT.87.) IA=1  
IF(FREQ.GT.82.) IB=1  
DO 100 K=1,6  
AA=AA+BSAT(K,IA)* $(FREQ^{**}(K-1))$   
BB=BB+BSBT(K,IB)* $(FREQ^{**}(K-1))$   
100 CONTINUE  
AA=EXP(AA)  
BSRAIN=AA*RAINRT**BB  
200 RETURN  
END  
MMR00710  
MMR00720  
MMR00730  
MMR00740  
MMR00750  
MMR00760  
MMR00770  
MMR00780  
MMR00790  
MMR00800
```

```

SUBROUTINE MMSNO(SNRT,TK,FQ,SNEX,SNBS) MMS00010
C----- MMS00020
PURPOSE: TO COMPUTE SNOW EXTINCTION AND BACKSCATTER X-SECTION. MMS00030
----- MMS00040
INPUTS: SNOWFALL RATE(NM/HR), TEMPERATURE(KELVIN), FREQUENCY(GHZ) MMS00050
----- MMS00060
OUTPUT: SNOW EXTINCTION (1/KM), BACKSCATTER(M**2/M**3). MMS00070
----- MMS00080
CALLED FROM HMMW. MMS00090
----- MMS00100
LOCAL VARIABLES: MMS00110
    FR, TS      LOCAL FREQUENCY, TEMPERATURE MMS00120
    ITYPE      INTEGER SNOW TYPE MMS00130
    XLMDA      WAVELENGTH(MMM) MMS00140
    FF0,1      FREQUENCY FITTING FACTORS MMS00150
    TF0,1      TEMPERATURE FITTING FACTORS MMS00160
    AA,10,1    INTERMEDIATE A VALUES MMS00170
    BB,10,1    INTERMEDIATE B VALUES MMS00180
    AA,BB      TERMS IN EXT(SNOW)=AA*SNOWRATE**BB MMS00190
    HBSLT      TERM IN BSCAT(SNOW)=AA*SNOWRATE**1.8 MMS00200
----- MMS00210
    AA,MENSH(A(9,3),B(9,3),F(9,3),T(3)),BSAT(6,2),BSBT(6,2),SFCT(3) MMS00220
    DATA AA/ 1.30E-3, 2.75E-3, 1.25E-2, 2.50E-2, 8.00E-2, 1.65E-1, MMS00230
        3.60E-1, 6.91E-1, 1.68E+0, MMS00240
        2.07E-2, 4.34E-2, 1.60E-1, 2.00E-1, 3.10E-1, 4.00E-1, MMS00250
        5.81E-1, 6.50E-1, 1.11E+0, MMS00260
        6.32E-2, 9.67E-2, 2.35E-1, 3.41E-1, 6.10E-1, 8.52E-1, MMS00270
        7.83E-1, 7.37E-1, 5.76E-1/ MMS00280
    DATA B/ 1.3, 1.46, 1.6, 1.54, 1.26, 1.1, .89, .79, .6, MMS00290
        1.3, 1.2, .95, .80, .75, .67, .65, .64, .60, MMS00300
        1.3, 1.2, .95, .80, .75, .67, .65, .64, .60/ MMS00310
    DATA F/ 10., 15., 35., 50., 95., 140., 225., 312., 1000./ MMS00320
    DATA T/ 271., 273., 275., SFCT/ 1., 3., 4.,/ MMS00330
    DATA BSAT/- .8824881E+01, -.1029998E-01, +.2451205E-04, MMS00340
        -.2462900E-07, +.6507628E-10, -.1856080E-12, MMS00350
        -.2127020E+02, +.6906017E+00, -.1924260E-01, MMS00360
        +.3035233E-03, -.2545323E-05, +.8673581E-08/, MMS00370
    DATA BSBT/ +.7901887E+00, -.1900189E-02, +.6341350E-05, MMS00380
        +.3186429E-08, -.5933950E-10, +.9056715E-13, MMS00390
        +.1361993E+01, +.3628100E-01, -.2461284E-02, MMS00400
        +.4805257E-04, -.3988064E-06, +.1193458E-08/ MMS00410
----- MMS00420
    TS=TK MMS00430
    FS=FQ MMS00440
    ICE=1 MMS00450
    SNEX=0, MMS00460
    SNBS=0, MMS00470
    IF(FQ.LT.10.,) GO TO 200 MMS00480
    XLMDA=299.79/FQ MMS00490
    CALL INTRP(A,B,T,FS,AA,BB,J) MMS00500
    SNEX=AA*SNRT**BB MMS00510
    SNBS=AA*BSAT(K,IA)*(FQ***(K-1)) MMS00520
    SNBS=BB*BSBT(K,IB)*(FQ***(K-1)) MMS00530
    CONTINUE MMS00540
    AA=EXP(AA) MMS00550
    BB=1.2*BB MMS00560
    F1=(FQ-10.)/85. MMS00570
    FCT=.367+F1*.633 MMS00580
----- MMS00590
    DO 100 K=1,6 MMS00600
    AA=AA+BSAT(K,IA)*(FQ***(K-1)) MMS00610
    BB=BB*BSBT(K,IB)*(FQ***(K-1)) MMS00620
    CONTINUE MMS00630
    AA=EXP(AA) MMS00640
    BB=1.2*BB MMS00650
    F1=(FQ-10.)/85. MMS00660
    FCT=.367+F1*.633 MMS00670
----- MMS00680
    100

```

IF(FQ.GT,.95.)	FCT=1.	MMS00710
AA=FCT*AA		MMS00720
110 IF(TK.GE.275.)	J=4	MMS00730
SNBS=SFCT(J-1)*AA*SNRT**BB		MMS00740
C 200 RETURN		MMS00750
C END		MMS00760
		MMS00770
		MMS00780

```

SUBROUTINE MMWFG(FD,T,FREQ,GFOG,BSFOG)          MMF00010
C*****                                         MMF00020
CALCULATES ABSORPTION DUE TO WATER FOGS/CLOUDS,   MMF00030
AND BACKSCATTER CROSS SECTION IN M**2/M**3.        MMF00040
INPUTS ARE: FOG DENSITY(GM/M**3), TEMPERTURE(KELVIN), MMF00050
FREQUENCY(GHZ).                                     MMF00060
OUTPUTS ARE: FOG ABSORPTION(/KM), BACKSCATTER X-SECTION(M**2 M**3). MMF00070
MMWFG IS CALLED FROM MMWMOD CALLS MMIDX SUBROUTINE. MMF00080
LOCAL VARIABLES:                                 MMF00090
XLMDA      WAVELENGTH(MM)                         MMF00100
C*****                                         MMF00110
ICE=0                                              MMF00120
IF(T.LT.243.) ICE=1                               MMF00130
C COMPUTE FOG EXTINCTION                         MMF00140
XLMDA=1.0./(FREQ/29.98)                           MMF00150
CALL MMIDX(XLMDA,T,ICE,H20AB,H20K2)               MMF00160
GFOG=18.8498*H20AB*FD/XLMDA                      MMF00170
BSFOG=1.162E-06*H20K2*FD**1.75/(XLMDA**4)       MMF00180
RETURN                                            MMF00190
C                                         MMF00200
END                                              MMF00210
                                         MMF00220
                                         MMF00230
                                         MMF00240
                                         MMF00250
                                         MMF00260
                                         MMF00270
}

```

```

SUBROUTINE MMWGS(T,P,AH,FREQ,GAS) MMG00010
***** MMG00020
SUBROUTINE COMPUTES GASEDUS ABSORPTION FROM 0 TO 1000 GHZ FOR MMG00030
H2O VAPOR AND O2. MMG00040
MMG00050
INPUTS INCLUDE: TEMPERATURE(KELVIN), PRESSURE(TORR), ABSOLUTE HUMIDITY MMG00060
IN GM/M**3, FREQUENCY(GIGAHERTZ). MMG00070
OUTPUTS ARE: GAS ABSORPTION (1/KM) MMG00080
MMG00090
MMG00100
MMWGS IS CALLED FROM NMMW, CALLS (1)MMOXY(OXYGEN ABSORPTION), MMG00110
(2) MMH2O(H2O VAPOR ABSORPTION). MMG00120
MMG00130
MMG00140
MMG00150
LOCAL VARIABLES: MMG00160
DATA02(L,J) : OXYGEN LINE DATA, L=LINE NUMBER, J=TYPE: MMG00170
    1 : LINE FREQUENCY(GHZ) MMG00180
    2 : LINE STRENGTH AT 300K MMG00190
    3 : LINE WIDTH AT 300K (GHZ/TORR) MMG00200
    4 : INTERFERENCE PARAMETER AT 300K MMG00210
    5 : INTERFERENCE TEMPERATURE CORRECTION MMG00220
    6 : LINE QUANTUM PARAMETER MMG00230
DATH2O(L, J) : H2O LINE DATA, L=LINE NUMBER, J=TYPE: MMG00240
    1 : WAVENUMBER(1/CM) MMG00250
    2 : STRENGTH MMG00260
    3 : WIDTH(1/CM/TORR) MMG00270
    4 : GROUND STATE ENERGY MMG00280
GH2O : H2O VAPOR ABSORPTION (1/KM) MMG00290
GU2 : O2 ABSORPTION (1/KM) MMG00300
PH2O : H2O VAPOR PRESSURE(TORR) MMG00310
WVNMB : WAVENUMBER(1/CM) MMG00320
MMG00330
DIMENSION DATA02(42,6), DATH2O(37,4) MMG00340
DIMENSION D102(42,3), D202(42,3), D1H2O(37,2), D2H2O(37,2) MMG00350
EQUIVALENCE (DATA02(1,1),D102(1,1)),(DATA02(1,4),D202(1,1)) MMG00360
EQUIVALENCE (DATH2O(1,1),D1H2O(1,1)),(DATH2O(1,3),D2H2O(1,1)) MMG00370
COMMON /IOUNIT/IOPN, IODIN, IODOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUM MG00380
C ##### DATA D102/ MMG00390
1 49.451,49.961,50.473,50.987,51.503,52.021,52.542,53.066, MMG00400
2 53.595,54.129,54.671,55.221,55.783,56.264,56.363,56.968, MMG00420
3 57.612,58.323,58.446,59.164,59.590,60.306,60.434,61.150, MMG00430
4 61.800,62.411,62.486,62.998,63.568,64.127,64.678,65.224, MMG00440
5 65.764,66.302,66.836,67.369,67.900,68.430,68.960,69.488, MMG00450
6 70.016,119.75, MMG00460
1 7.E-5,2.2E-4,6.E-4,1.56E-3,3.86E-3,8.99E-3,1.971E-2,.04072, MMG00470
2 .07919,.1448,.2489,.4012,.6056,.3487,.8539,1.1204,1.3595, MMG00480
3 1.515,.9251,1.5263,1.341,1.3487,1.5626,1.5899,1.4588,1.2272, MMG00490
4 .9634,.954,.6898,.4656,.2942,.1744,.0971,.0508,.025,.0116, MMG00500
5 5.08E-3,2.1E-3,8.2E-4,3.E-4,1.E-4,.5973, MMG00510
1 1.260,1.310,1.330,1.360,1.380,1.410,1.440,1.460,1.490, MMG00520
2 1.510,1.540,1.570,1.601,2.212,1.635,1.672,1.714,1.762, MMG00530
3 1.964,1.819,1.859,1.890,1.789,1.736,1.694,1.658,1.990, MMG00540
4 1.627,1.598,1.568,1.540,1.510,1.490,1.460,1.440,1.410, MMG00550
5 1.380,1.360,1.330,1.310,1.280,2.140/, MMG00560
DATA D202/ MMG00570
1 0.000,0.000,0.000,1.040,0.802,0.897,0.825,0.780,0.764, MMG00580
2 0.666,0.651,0.550,0.481,0.931,0.371,0.254,0.100,-.087, MMG00590
3 0.729,-.318,0.433,-.543,0.179,-.028,-.183,-.324,-.615, MMG00600
4 -.419,-.537,-.591,-.693,-.703,-.796,-.808,-.849,-.916, MMG00610
5 -.822,-1.05,0.000,0.000,0.000,-.054, MMG00620
1 1.00,1.00,1.00,1.38,2.04,1.69,1.91,1.88,1.90,2.01,1.95, MMG00630
2 2.11,2.13,0.89,2.36,2.66,4.20,-5.8,0.79,0.11,0.50,0.69, MMG00640
3 -.99,7.60,3.04,2.34,0.85,2.24,2.02,2.04,1.89,1.95,1.85, MMG00650
4 1.83,1.86,1.66,1.99,1.36,1.00,1.00,1.00,0.89, MMG00660
1 41.,39.,37.,35.,33.,31.,29.,27.,25.,23.,21.,19.,17.,1.,15.,13., MMG00670
2 11.,9.,3.,7.,5.,5.,7.,9.,11.,13.,3.,15.,17.,19.,21.,23.,25., MMG00680
3 27.,29.,31.,33.,35.,37.,39.,41.,1./ MMG00690
##### MMG00700

```

```

DATA D1H20/
1 00.742,.06.115,.06.790,10.715,10.846,12.682,14.778,14.944, MMG00710
2 15.707,15.834,16.294,18.270,18.577,20.704,21.960,24.860, MMG00720
3 25.085,30.560,32.366,32.954,36.604,37.137,37.910,38.638, MMG00730
4 38.791,39.112,40.282,40.520,42.639,43.243,43.631,44.100, MMG00740
5 46.750,47.053,48.059,49.765,49.820, MMG00750
1 .436E-24, .775E-22, .186E-24, .250E-23, .906E-22, .827E-21, MMG00760
2 .145E-22, .863E-21, .270E-22, .108E-21, .219E-22, .985E-22, MMG00770
3 .526E-19, .565E-21, .531E-22, .664E-22, .347E-19, .143E-20, MMG00780
4 .160E-20, .252E-19, .164E-18, .502E-19, .333E-21, .242E-20, MMG00790
5 .179E-18, .197E-21, .558E-19, .155E-21, .707E-21, .687E-21, MMG00800
6 .511E-22, .568E-20, .302E-21, .142E-18, .930E-21, .399E-22, MMG00810
7 .478E-22/ MMG00820
DATA D2H20/
1 .081, .094, .095, .063, .087, .091, .050, .083, .061, .071, .075, MMG00830
2 .111, .107, .072, .111, .103, .102, .084, .083, .101, .097, .099, MMG00840
3 .094, .073, .094, .063, .093, .098, .066, .074, .060, .081, .084, MMG00850
4 .091, .078, .096, .097, MMG00860
1 446.512, 136.164, 134.800, 1284.921, 315.780, 212.156, MMG00870
2 1045.069, 285.419, 742.079, 488.136, 586.482, 23.750, MMG00880
3 23.794, 488.110, 1618.550, 69.920, 70.091, 285.219, MMG00890
4 383.843, 37.137, 136.761, 0.000, 172.880, 610.345, MMG00900
5 173.366, 888.641, 275.498, 1731.890, 888.607, 842.361, MMG00910
6 1079.088, 508.814, 398.392, 399.459, 601.553, 100.391, MMG00920
7 1693.650/ MMG00930
C#####
C COMPUTE WATER VAPOR PRESSURE, FREQUENCY IN WAVENUMBERS MMG00940
PH20=AH*T*3.462977E-3 MMG00950
IF(AH.LT.0.) PH20=-PSAT(T)*AH/100. MMG00960
WVNMB=FRQ/29.98 MMG00970
C COMPUTE H2O ABSORPTION MMG00980
CALL MMH20(WVNMB,T,P,PH20,DATH20,GH20) MMG01000
C COMPUTE O2 ABSORPTION MMG01010
IF(FRQ.LT.140.) CALL MMOXY(FRQ,T,P,PH20,DATA02,G02) MMG01020
C SUM ABSORPTION MMG01030
GAS=GH20+G02 MMG01040
C***** MMG01050
RETURN MMG01060
END MMG01070

```

```

C      FUNCTION PFR(T)
C      COMPUTE H2O PARTITION FUNCTION CORRECTIONS
C
C      DIMENSION VIB(3)
C      DATA VIB/3693.9,1614.5,3801.8/
C
C      QJ=296./T
C      QJ=QJ*SQRT(QJ)
C      T1=-1.43879/296.
C      T2=-1.43879/T
C      T1S=1.
C      T2S=1.
C
C      DO 10 J=1,3
C          V=VIB(J)
C          T11=1.-EXP(-T1*V)
C          T22=1.-EXP(-T2*V)
C          T1S=T1S*T11
C          T2S=T2S*T22
C          PFR=QJ*T2S/T1S
C
C      RETURN
C
C      END

```

PFR00010
 PFR00020
 PFR00030
 PFR00040
 PFR00050
 PFR00060
 PFR00070
 PFR00080
 PFR00090
 PFR00100
 PFR00110
 PFR00120
 PFR00130
 PFR00140
 PFR00150
 PFR00160
 PFR00170
 PFR00180
 PFR00190
 PFR00200
 PFR00210
 PFR00220
 PFR00230

```

C FUNCTION PSAT(T)                                PSAT0010
DATA C1,C2,C3,C4,C5,C6,C7/-7.90298,5.02808,-1.3816E-7,11.344,    PSAT0020
+ 8.1328E-3,-3.49149,3.005715/                  PSAT0030
DATA D1,D2,D3,D4/-9.09718,-3.56654,.876793,.785835/                 PSAT0040
DATA TS,T0/373.16,273.16/,CONV/.7500646/          PSAT0050
IF(T.LE.T0) GO TO 100                           PSAT0060
TR=TS/T                                         PSAT0070
TRI=T/TS                                         PSAT0080
EW=C1*(TR-1.)+C2*ALOG10(TR)+C3*(10.**C4*(1.-TR))-1.)+    PSAT0090
+ C5*(10.**C6*(TR-1.))-1.)+C7                  PSAT0100
GO TO 200                                         PSAT0110
TR=T0/T                                         PSAT0120
TRI=T/T0                                         PSAT0130
EW=D1*(TR-1.)+D2*ALOG10(TR)+D3*(1.-TR)+D4        PSAT0140
PSAT=(10.**EW)*CONV                         PSAT0150
RETURN                                           PSAT0160
END                                              PSAT0170
                                                PSAT0180

```

```

C      FUNCTION SUPK(A,B,C)
C      COMPUTES THE SUPER KINETIC LINE PROFILE FACTOR.
C
C      PI=3.14159265
C      XNORM=.998776
C      VM=10.*C
C      X=ABS((B*B-A*A)/(2.*C*C))
C      IF(ABS(B-A).GT.VM) GO TO 10
C
C      SUPK=XNORM/(PI*C)/(X*X+1.)
C      RETURN
C
C10    CXI=10.*((B+10.*C/2.)/(B+10.*C))
C      CXI=((CXI**1.88)+1.)/((CXI*CXI)+1.)
C      SUPK=XNORM*CXI/(PI*C)/(X*X+1.)
C      RETURN
C      END

```

SUPK0010
SUPK0020
SUPK0030
SUPK0040
SUPK0050
SUPK0060
SUPK0070
SUPK0080
SUPK0090
SUPK0100
SUPK0110
SUPK0120
SUPK0130
SUPK0140
SUPK0150
SUPK0160
SUPK0170

```

SUBROUTINE CLTRAN(CTRANS,WAVE,IRUN,IERR) CLT00010
DIMENSION TAU(20) CLT00020
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20) CLT00030
COMMON /CLYMAT/TTEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
+ WNDVEL,WNDDIR,IPASCT
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT CLT00040
COMMON /PATHL/X0,Y0,Z0,X1,Y1,XS,YS,ZS,XT,YT,ZT,ATTL(20) CLT00050
COMMON /BASTOP/ZBAS,ZTOP,ICL,INW,ILCTYP(10),ICCTYP(10) CLT00060
COMMON /CYL/XC,YC,RADIUS CLT00070
COMMON /IQUINT/IQIN,IOIN,IPHFUN,LOUNIT,HDIRTU,HCLIMT,KSTOR,NPLCTU CLT00080
COMMON /BASTH/ZLBASE(10),ZLTHICK(10),ZCBASE(10),ZCTHICK(10) CLT00090
+ RADICL(10) CLT00100
COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10) CLT00110
CLT00120
C-- DATA IS READ FROM INPUT RECORDS AND THEN TRANSFERRED TO CLTRAN CLT00130
C** BY SUBROUTINE CLREAD. INPUT TO CLTRAN IS CARD ORDER-INDEPENDENT, CLT00140
C** WITH A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH INPUT RECORD. CLT00150
C** THE ONLY EXCEPTION TO THIS ORDER-INDEPENDENCE IS THE GO SENTINEL CLT00160
C** CARD, WHICH MUST BE THE LAST RECORD READ. ALL CARDS ARE READ IN CLT00170
C** UNDER THE FORMAT (A4,1X,5E10.5,1X). THE IDENTIFICATION AND MEANING CLT00180
C** OF EACH INPUT RECORD ARE AS FOLLOWS : CLT00190
CLT00200
CARD IDENTIFIER : SEEK CLT00210
VARIABLES READ : XS,YS,ZS CLT00220
XS,YS,ZS = POSITION COORDINATES OF SEEKER (OR RECEIVER) (KM) CLT00230
CLT00240
CARD IDENTIFIER : TARG CLT00250
VARIABLES READ : XT,YT,ZT CLT00260
XT,YT,ZT = POSITION COORDINATES OF TARGET (KM) CLT00270
CLT00280
CLT00290
C-- THE FOLLOWING FOUR CARDS REPRESENT THE STRATIFORM CLOUDS TREATED CLT00300
C** BY CLTRAN. THEY ARE TREATED AS INFINITE LAYERS IN A PLANE-PARALLEL CLT00310
C** ATMOSPHERE. CLT00320
CLT00330
CLT00340
CARD IDENTIFIER : CLST (STRATUS CLOUD TYPE) CLT00350
VARIABLES READ : ZLBASE,ZLTHIC CLT00360
ZLBASE = HEIGHT OF CLOUD BASE (KM) CLT00370
ZLTHIC = VERTICAL THICKNESS OF CLOUD LAYER (KM) CLT00380
CLT00390
CARD IDENTIFIER : CLAS (ALTOSTRATUS CLOUD TYPE) CLT00400
VARIABLES READ : ZLBASE,ZLTHIC CLT00410
CLT00420
CARD IDENTIFIER : CLNS (NIMBOSTRATUS CLOUD TYPE) CLT00430
VARIABLES READ : ZLBASE,ZLTHIC CLT00440
CLT00450
CARD IDENTIFIER : CLSC (STRATOCUMULUS CLOUD TYPE) CLT00460
VARIABLES READ : ZLBASE,ZLTHIC CLT00470
CLT00480
CLT00490
C-- THE NEXT TWO CARDS REPRESENT THE CUMULUS CLOUD TYPES ADDRESSED CLT00500
C** BY CLTRAN. CLOUDS OF THIS KIND ARE MODELLED AS CYLINDERS WHICH CLT00510
C** HAVE VERTICAL SYMMETRY AXES. THESE TYPES ARE REPRESENTED BY CLT00520
C** THE FOLLOWING RECORDS : CLT00530
CLT00540
CLT00550
CARD IDENTIFIER : CLCH (CUMULUS HUMILIS CLOUD TYPE) CLT00560
VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD CLT00570
ZCBASE = HEIGHT OF CLOUD CYLINDER'S LOWER BASE (KM) CLT00580
ZCTHIC = VERTICAL THICKNESS OF CLOUD CYLINDER (KM) CLT00590
RADICL = RADIUS OF CLOUD CYLINDER (KM) CLT00600
XCLOUD = X-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM) CLT00610
YCLOUD = Y-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM) CLT00620
CLT00630
CARD IDENTIFIER : CLCC (CUMULUS CONGESTUS CLOUD TYPE) CLT00640
VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD CLT00650
CLT00660
CLT00670
C-- THE FOLLOWING CARD MUST BE THE LAST RECORD READ : CLT00680

```

```

C----- CLT00690
C----- CLT00700
C----- CARD IDENTIFIER : GO CLT00710
C----- VARIABLES READ : NONE CLT00720
C----- CLT00730
C----- CLT00740
C----- IF(IRUN.GT.1) GO TO 5 CLT00750
C----- NLAYER=0 CLT00760
C----- NCCLDS=0 CLT00770
C----- CLT00780
5 CONTINUE
IWV=0
ZBMIN=999.
FOGPRB=0.
IF(WAVE.GT.0.20.AND.WAVE.LT.2.00) IWV=1
IF(WAVE.GT.3.00.AND.WAVE.LT.5.00) IWV=2
IF(WAVE.GT.8.0.AND.WAVE.LT.12.0) IWV=3
IF(IWV.NE.0) GO TO 50
IERR=1
IWRTEC(1D0UT,22)
22 FORMAT(1H0,20X,94H***CLTRAN ERROR*** INPUT WAVELENGTH DOES NOT LIE
+ WITHIN ALLOWABLE LIMITS, EXECUTION TERMINATED //)
GO TO 900
50 CALL CLREAD(NLAYER,NCCLDS,IERR)
IF(IERR.EQ.1) GO TO 900
ISLTUP=0
ISLTDN=0
IHORIZ=0
IVERT=0
C***** DETERMINE SENSE OF L-O-S SLOPE FROM SEEKER'S POINT OF VIEW
C----- CLT00940
C----- TESVER=(XS-XT)**2+(YS-YT)**2 CLT00950
C----- IF(TESVER.EQ.0.0)IVERT=1 CLT00960
C----- IF(IVERT.EQ.1)GO TO 300 CLT00970
C----- IF(2S-ZT)>200,210,220 CLT00980
200 ISLTUP=1
GO TO 300
210 IHORIZ=1
GO TO 300
220 ISLTDN=1
300 CONTINUE
IF(IHORIZ.EQ.1)GO TO 310
C***** COMPUTE L-O-S SLOPES IN X-Z AND Y-Z VERTICAL PLANES
C----- CLT01000
C----- XIX=(XS-XT)/(ZS-ZT) CLT01010
C----- XIY=(YS-YT)/(ZS-ZT) CLT01020
310 CONTINUE
C***** STRATIFORM CLOUD BLOCK
C----- CLT01030
C----- IF(NLAYER.EQ.0)GO TO 500 CLT01040
C***** UTILIZE DEFAULT BASE OR THICKNESS VALUES IF NECESSARY
C----- CLT01050
C----- CALL DEFSET(1,NLAYER)
DO 400 N=1,NLAYER
NLINT(N)=0
Z0(N)=ZLBASE(N)
Z1(N)=Z0(N)+ZLTHIC(N)
ZBAS=Z0(N)
ZTOP=Z1(N)
ICL=ILCTYP(N)
IF(ZBMIN.LT.ZLBASE(N)) GO TO 320
ZBMIN=ZLBASE(N)
FOGPRB=FLGAT(ICL)
320 CONTINUE
C***** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CLOUD LAYER (IF
C***** THERE ARE ANY); (X1(N),Y1(N),Z1(N))= UPPER INTERSECTION POINT,
C***** (X0(N),Y0(N),Z0(N))= LOWER INTERSECTION POINT
CLT01290
CLT01300
CLT01310
CLT01320

```

```

C          CALL LAYRXY(XIX,XIY,N)           CLT01330
C          IF(TESALL.LE.0.0)GO TO 350       CLT01340
C          NLINT(N)=N                     CLT01350
C          NLINT(N)=N                     CLT01360
C          NLINT(N)=N                     CLT01370
C          NLINT(N)=N                     CLT01380
C          NLINT(N)=N                     CLT01390
C          NLINT(N)=N                     CLT01400
C          NLINT(N)=N                     CLT01410
C          NLINT(N)=N                     CLT01420
C          NLINT(N)=N                     CLT01430
C          NLINT(N)=N                     CLT01440
C          NLINT(N)=N                     CLT01450
C          NLINT(N)=N                     CLT01460
C          NLINT(N)=N                     CLT01470
C          NLINT(N)=N                     CLT01480
C          NLINT(N)=N                     CLT01490
C          NLINT(N)=N                     CLT01500
C          NLINT(N)=N                     CLT01510
C          NLINT(N)=N                     CLT01520
C          NLINT(N)=N                     CLT01530
C          NLINT(N)=N                     CLT01540
C          NLINT(N)=N                     CLT01550
C          NLINT(N)=N                     CLT01560
C          NLINT(N)=N                     CLT01570
C          NLINT(N)=N                     CLT01580
C          NLINT(N)=N                     CLT01590
C          NLINT(N)=N                     CLT01600
C          NLINT(N)=N                     CLT01610
C          NLINT(N)=N                     CLT01620
C          NLINT(N)=N                     CLT01630
C          NLINT(N)=N                     CLT01640
C          NLINT(N)=N                     CLT01650
C          NLINT(N)=N                     CLT01660
C          NLINT(N)=N                     CLT01670
C          NLINT(N)=N                     CLT01680
C          NLINT(N)=N                     CLT01690
C          NLINT(N)=N                     CLT01700
C          NLINT(N)=N                     CLT01710
C          NLINT(N)=N                     CLT01720
C          NLINT(N)=N                     CLT01730
C          NLINT(N)=N                     CLT01740
C          NLINT(N)=N                     CLT01750
C          NLINT(N)=N                     CLT01760
C          NLINT(N)=N                     CLT01770
C          NLINT(N)=N                     CLT01780
C          NLINT(N)=N                     CLT01790
C          NLINT(N)=N                     CLT01800
C          NLINT(N)=N                     CLT01810
C          NLINT(N)=N                     CLT01820
C          NLINT(N)=N                     CLT01830
C          NLINT(N)=N                     CLT01840
C          NLINT(N)=N                     CLT01850
C          NLINT(N)=N                     CLT01860
C          NLINT(N)=N                     CLT01870
C          NLINT(N)=N                     CLT01880
C          NLINT(N)=N                     CLT01890
C          NLINT(N)=N                     CLT01900
C          NLINT(N)=N                     CLT01910
C          NLINT(N)=N                     CLT01920
C          NLINT(N)=N                     CLT01930
C          NLINT(N)=N                     CLT01940
C          NLINT(N)=N                     CLT01950
C          NLINT(N)=N                     CLT01960
C          NLINT(N)=N                     CLT01970
C          NLINT(N)=N                     CLT01980
C
C***** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH
C
C          CALL CLEXTN(TAUN,N)
C          TAUN(N)=TAUN
C          ATT(N)=SQRT((X1(N)-X0(N))**2+(Y1(N)-Y0(N))**2+
C          +(Z1(N)-Z0(N))**2)
C          GO TO 400
C
C***** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH
C***** IS NOW SET TO ZERO
C
C          350 TAUN(N)=0.0
C          ATT(N)=0.0
C          400 CONTINUE
C
C***** END STRATIFORM CLOUD BLOCK
C
C***** CUMULIFORM CLOUD BLOCK
C
C          500 IF(NCCLDS.EQ.0)GO TO 805
C
C***** UTILIZE DEFAULT BASE, THICKNESS, OR RADIUS IF NECESSARY
C
C          CALL DEFSET(2,NCCLDS)
C          DO 800 N=1,NCCLDS
C          NN=NAYER+N
C          NCINT(N)=0
C          Z0(NN)=ZCBASE(N)
C          Z1(NN)=Z0(NN)+ZCTHIC(N)
C          ICL=ICCTYP(N)
C          IF(ZBMIN.LT.ZCBASE(N)) GO TO 520
C          ZBMIN=ZCBASE(N)
C          FOGPRB=FLOAT(ICL)
C
C          520 CONTINUE
C          RADIUS=RADIICL(N)
C          XC=XCLOUD(N)
C          YC=YCLOUD(N)
C          ZBAS=Z0(NN)
C          ZTOF=Z1(NN)
C          TESARG=0.0
C          TESVT=0.0
C
C***** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CUMULUS CLOUD CYLINDER
C***** (IF THERE ARE ANY); (X1(N),Y1(N),Z1(N))= UPPER INTERSECTION POINT,
C***** (X0(N),Y0(N),Z0(N))= LOWER INTERSECTION POINT
C
C          CALL CYLXY(XIX,XIY,NN)
C          IF(TESALL.LE.0.0.OR.TESARG.LT.0.0.OR.TESVT.LT.0.0)GO TO 650
C          NCINT(N)=N
C
C***** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH
C
C          CALL CLEXTN(TAUN,NN)
C          TAUN(NN)=TAUN
C          ATT(NN)=SQRT((X1(NN)-X0(NN))**2+(Y1(NN)-Y0(NN))**2+
C          +(Z1(NN)-Z0(NN))**2)
C          GO TO 800
C
C***** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH
C***** IS NOW SET TO ZERO
C
C          650 TAUN(NN)=0.0
C          ATT(NN)=0.0
C          800 CONTINUE
C
C***** END CUMULIFORM CLOUD BLOCK

```

```
C 805 CONTINUE
C**** DETERMINE CUMULATIVE CLOUD OPTICAL DEPTH AND TRANSMITTANCE
TAUTOT=0,0
NN=NAYER+NCCLDS
IF(NN.EQ.0)GO TO 820
DO 810 N=1,NN
810 TAUTOT=TAUTOT+TAU(N)
820 CTRANS=EXP(-TAUTOT)
850 CALL LISOUT(NN,CTRANS,TAUTOT,TAU,NAYER,IRUN)
900 RETURN
END
```

```
CLT01990
CLT02000
CLT02010
CLT02020
CLT02030
CLT02040
CLT02050
CLT02060
CLT02070
CLT02080
CLT02090
CLT02100
```

```

SUBROUTINE LAYRXY(XIX,XIY,IN)                               LAY00010
LOGICAL INSEEK,INTARG                                      LAY00020
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)      LAY00030
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT LAY00040
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20) LAY00050
C**** CHECK LOS TO SEE IF IT INTERSECTS THE CLOUD LAYER      LAY00060
UP1=ISLTUP*(ZS-Z1(IN))                                     LAY00070
UP2=ISLTUP*(Z0(IN)-ZT)                                     LAY00080
DN1=ISLTDN*(ZT-Z1(IN))                                     LAY00090
DN2=ISLTDN*(Z0(IN)-ZS)                                     LAY00100
HOR12=IHORIZ2*(ZS-Z1(IN))*((ZS-Z0(IN))*(-1,0))          LAY00110
UPVER1=IVERT*((ZT-ZS)*(ZS-Z1(IN)))                         LAY00120
UPVER2=IVERT*((ZT-ZS)*(Z0(IN)-ZT))                         LAY00130
TESALL=UP1*UP2+DN1*DN2+HOR12+UPVER1*UPVER2                LAY00140
IF(TESALL.LT.0.0)RETURN                                     LAY00150
IF(IHORIZ.EQ.1)RETURN                                     LAY00160
C
C**** COMPUTE X,Y INTERSECTIONS OF CLOUD PLANES AND LOS      LAY00170
C**** ALSO, CHECK FOR THE CASE WHERE EITHER THE SEEKER       LAY00180
C**** OR TARGET IS INSIDE OF THE CLOUD                      LAY00190
C
INSEEK=ZS.LE.Z1(IN).AND.ZS.GE.Z0(IN)                      LAY00200
INTARG=ZT.LE.Z1(IN).AND.ZT.GE.Z0(IN)                      LAY00210
IF(ISLTUP.EQ.1.AND.INTARG.GT1(IN)=ZT                     LAY00220
IF(ISLTDN.EQ.1.AND.INTARG.GT0(IN)=ZT                     LAY00230
IF(ISLTUP.EQ.1.AND.INSEEK.GT0(IN)=ZS                     LAY00240
IF(ISLTDN.EQ.1.AND.INSEEK.GT1(IN)=ZS                     LAY00250
IF(INSEEK=Z1(IN)=XS+XI*X*(Z1(IN)-ZS)                   LAY00260
Y1(IN)=YS+XI*Y*(Z1(IN)-ZS)                             LAY00270
X0(IN)=XS+XI*X*(Z0(IN)-ZS)                             LAY00280
Y0(IN)=YS+XI*Y*(Z0(IN)-ZS)                             LAY00290
RETURN
END

```

```

SUBROUTINE CYLXY(XIX,XIY,IN) CYL00010
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20) CYL00020
COMMON /CYL/XC,YC,RADIUS CYL00030
COMMON /LEVEL/ISLTOP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT CYL00040
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20) CYL00050
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10) CYL00060
***** FIRST DETERMINE INTERSECTION PTS OF LOS WITH UPPER CYL00070
***** AND LOWER BASE PLANES OF CLOUD CYLINDER CYL00080
TESALL=0.0 CYL00090
CALL LAYRXY(XIX,XIY,IN) CYL00100
IF(TESALL.LT.0.0)RETURN CYL00110
X0LAYR=X0(IN) CYL00120
Y0LAYR=Y0(IN) CYL00130
Z0LAYR=Z0(IN) CYL00140
X1LAYR=X1(IN) CYL00150
Y1LAYR=Y1(IN) CYL00160
Z1LAYR=Z1(IN) CYL00170
***** CALCULATE NEXT THE INTERSECTION PTS OF THE LOS CYL00180
***** WITH THE SURFACE OF AN INFINITE VERTICAL CYLINDER WITH CYL00190
***** THE SAME RADIUS (RADIUS)AND LATERAL POSITION (XC,YC) AS THE CLOUD CYL00200
IF(IVERT.NE.1)GO TO 40 CYL00210
VERTES=SQRT((XT-XC)**2+(YT-YC)**2) CYL00220
TESVT=RADIUS-VERTES CYL00230
IF(VERTES.LT.RADIUS)GO TO 400 CYL00240
RETURN CYL00250
40 IF((XT-XS).EQ.0.0)GO TO 60 CYL00260
ALPHA=(YT-YS)/(XT-XS) CYL00270
A2=ALPHA**2 CYL00280
C2=1.0+A2 CYL00290
C1=0.0*(ALPHA*(YT-YC-ALPHA*XT)-XC) CYL00300
C0=XC**2+(YT-YC-ALPHA*XT)**2-RADIUS**2 CYL00310
TESARG=C1**2-4.0*C2*C0 CYL00320
IF(TESARG.LT.0.0)RETURN CYL00330
XP=(-C1+SQRT(TESARG))/(2.0*C2) CYL00340
XM=(-C1-SQRT(TESARG))/(2.0*C2) CYL00350
YP=YT+ALPHA*(XP-XT) CYL00360
YM=YT+ALPHA*(XM-XT) CYL00370
IF(IHORIZ.EQ.1)GO TO 300 CYL00380
ZM=ZS+(1.0/XIX)*(XM-XS) CYL00390
ZP=ZS+(1.0/XIX)*(XP-XS) CYL00400
***** CHECK FOR SKEW MISS OF CLOUD CYL00410
TOPP=ZP-ZTOP CYL00420
TOPM=ZM-ZTOP CYL00430
BASP=ZBAS-ZP CYL00440
BASM=ZBAS-ZM CYL00450
IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50 CYL00460
TESALL=-1.0 CYL00470
RETURN CYL00480
60 XP=XT CYL00490
TESARG=RADIUS**2-(XP-XT)**2 CYL00500
IF(TESARG.LT.0.0)RETURN CYL00510
YP=YC+SQRT(TESARG) CYL00520
XM=XT CYL00530
YM=YC+SQRT(TESARG) CYL00540
IF(IHORIZ.EQ.1)GO TO 300 CYL00550
ZM=ZS+(1.0/XIY)*(YM-YS) CYL00560
ZP=ZS+(1.0/XIY)*(YP-YS) CYL00570
***** CHECK FOR SKEW MISS OF CLOUD CYL00580
TOPP=ZP-ZTOP CYL00590
TOPM=ZM-ZTOP CYL00600
BASP=ZBAS-ZP CYL00610
BASM=ZBAS-ZM CYL00620
IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50 CYL00630
TESALL=-1.0 CYL00640
RETURN CYL00650
50 IF(ZP.LT.ZM)GO TO 100 CYL00660
X0CYL=XM CYL00670
Y0CYL=YM CYL00680
Z0CYL=ZM CYL00690
X1CYL=XP CYL00700

```

	Y1CYL=YP	CYL00710
	Z1CYL=ZP	CYL00720
	GO TO 200	CYL00730
100	X0CYL=XP	CYL00740
	Y0CYL=YP	CYL00750
	Z0CYL=ZP	CYL00760
	X1CYL=XN	CYL00770
	Y1CYL=YM	CYL00780
	Z1CYL=ZN	CYL00790
200	CONTINUE	CYL00800
	GO TO 450	CYL00810
300	X0<IN>=XM	CYL00820
	Y0<IN>=YM	CYL00830
	Z0<IN>=ZT	CYL00840
	X1<IN>=XP	CYL00850
	Y1<IN>=YP	CYL00860
	Z1<IN>=ZT	CYL00870
	GO TO 500	CYL00880
400	X0<IN>=XT	CYL00890
	Y0<IN>=YT	CYL00900
	X1<IN>=XT	CYL00910
	Y1<IN>=YT	CYL00920
	GO TO 500	CYL00930
450	IF<Z0CYL,LT,Z0LAYR>GO TO 460	CYL00940
	X0<IN>=X0CYL	CYL00950
	Y0<IN>=Y0CYL	CYL00960
	Z0<IN>=Z0CYL	CYL00970
	GO TO 470	CYL00980
460	X0<IN>=X0LAYR	CYL00990
	Y0<IN>=Y0LAYR	CYL01000
	Z0<IN>=Z0LAYR	CYL01010
470	IF<Z1CYL,GT,Z1LAYR>GO TO 480	CYL01020
	X1<IN>=X1CYL	CYL01030
	Y1<IN>=Y1CYL	CYL01040
	Z1<IN>=Z1CYL	CYL01050
	GO TO 500	CYL01060
480	X1<IN>=X1LAYR	CYL01070
	Y1<IN>=Y1LAYR	CYL01080
	Z1<IN>=Z1LAYR	CYL01090
500	RETURN	CYL01100
	END	CYL01110

```

SUBROUTINE CLEXTN(TAUN, IN) CLE00010
DIMENSION AA(42), BB(42), CC(42), A(7,6,3) CLE00020
DIMENSION X1(20), Y1(20), Z1(20), X0(20), Y0(20), Z0(20) CLE00030
EQUIVALENCE (AA(1), A(1,1,1)), (BB(1), A(1,1,2)), (CC(1), A(1,1,3)) CLE00040
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20) CLE00050
COMMON /LEVEL/ISLTUP, ISLTDN, IHORIZ, IVERT, TESALL, TESARG, TESVT CLE00060
COMMON /BASTOP/ZBAS, ZTOP, ICL, IWY CLE00070
DATA AA/5.3744E-2, 9.313E-4, -1.1300E-6, 1.6228E-9, -1.0421E-12, CLE00080
+2.4192E-16, 0.0000E-00, 8.4451E-2, 1.6549E-4, -4.7300E-7, CLE00090
+4.4334E-10, -1.3851E-13, -2.0644E-17, 1.2569E-20, 1.0240E-1, CLE00100
+4.8678E-5, -2.2195E-7, 1.4506E-10, 5.9393E-14, -8.7250E-17, CLE00110
+2.1383E-20, 3.6775E-2, 1.6160E-4, -6.4666E-7, 9.5298E-10, CLE00120
+-6.4023E-13, 1.6916E-16, -7.9924E-21, 1.8410E-2, 6.6870E-4, CLE00130
+-2.8406E-6, 5.4478E-9, -5.4854E-12, 2.7932E-15, -5.8271E-19, CLE00140
+3.3458E-2, 1.3098E-4, -4.1528E-7, 6.1166E-10, -4.6681E-13, CLE00150
+1.7383E-16, -2.5043E-20/ CLE00160
DATA BB/7.5099E-2, 3.2061E-4, -1.7060E-6, 3.3538E-9, -3.2719E-12, CLE00170
+1.5721E-15, -3.0123E-19, 1.1808E-1, 2.2387E-4, -8.7996E-7, CLE00180
+1.1298E-9, -6.6823E-13, 1.7514E-16, -1.5665E-20, 1.4155E-1, CLE00190
+-5.0592E-5, -1.2280E-7, 1.2715E-10, 2.7470E-14, -6.6940E-17, CLE00200
+1.7734E-20, 4.9533E-2, 2.0904E-4, -1.1626E-6, 2.3531E-9, CLE00210
+-2.3463E-12, 1.1472E-15, -2.2267E-19, 3.8315E-3, 1.1837E-3, CLE00220
+-5.1096E-6, 9.4660E-9, -8.6426E-12, 3.6778E-15, -5.5775E-19, CLE00230
+4.1534E-2, 2.0220E-4, -8.1463E-7, 1.3590E-9, -1.1142E-12, CLE00240
+4.3867E-16, -6.6426E-20/ CLE00250
DATA CC/1.1269E-2, 2.8659E-4, -6.0210E-8, -1.5274E-9, 2.7747E-12, CLE00260
+-1.8946E-15, 4.5539E-19, 1.9856E-2, 1.2292E-4, -8.4479E-8, CLE00270
+8.6563E-12, 0.0000E-0, 0.0000E-0, 0.0000E-0, 3.1907E-2, CLE00280
+1.9632E-4, -1.1114E-7, -3.3987E-10, 5.2528E-13, -2.7799E-16, CLE00290
+5.0469E-20, 6.8522E-3, 1.5362E-4, 7.5813E-8, -1.1430E-9, CLE00300
+1.8885E-12, -1.2477E-15, 2.9522E-19, 8.5792E-4, 6.4122E-5, CLE00310
+7.7271E-7, -2.9750E-9, 4.4014E-12, -3.0626E-15, 8.1541E-19, CLE00320
+3.7151E-3, 1.4919E-4, -9.3486E-8, -1.2183E-10, 1.8733E-13, CLE00330
+-9.2167E-17, 1.5939E-20/ CLE00340
TAUN=0.0 CLE00350
ZZA=Z0(IN)-ZBAS CLE00360
ZZB=Z1(IN)-ZBAS CLE00370
VERDIS=ZZB-ZZA CLE00380
HORDIS=SQRT((X1(IN)-X0(IN))**2+(Y1(IN)-Y0(IN))**2) CLE00390
EL=SQRT(HORDIS**2+VERDIS**2) CLE00400
IF(EL.EQ.0.0)RETURN CLE00410
IF(IHORIZ.EQ.1)GO TO 200 CLE00420
XI=VERDIS/EL CLE00430
ELA=ZZA*1000.0 CLE00440
ELB=(XI*EL+ZZA)*1000.0 CLE00450
POLYA=0.0 CLE00460
POLYB=0.0 CLE00470
DO 100 N=1,? CLE00480
EN=FLOAT(N) CLE00490
AN=A(N, ICL, IWY) CLE00500
TERMB=(1.0/(XI*EN))*AN*ELB**N CLE00510
TERMA=(1.0/(XI*EN))*AN*ELA**N CLE00520
POLYA=POLYA+TERMA CLE00530
100 POLYB=POLYB+TERMB CLE00540
TAUN=POLYB-POLYA CLE00550
GO TO 300 CLE00560
200 ZZA=ZZA*1000.0 CLE00570
DO 250 N=1,? CLE00580
TERMK=A(N, ICL, IWY)*HORDIS*1000.0*ZZA**N CLE00590
250 TAUN=TAUN+TERMK CLE00600
300 RETURN CLE00610
END CLE00620

```

```

C SUBROUTINE LISOUT(NN,CTRANS,TAUTOT,TAU,NLAYER,IRUN) LIS00010
C***** OUTPUT CONTROL ROUTINE LIS00020
C
C DIMENSION ALPH(6),T(20),TAU(20) LIS00030
C DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20) LIS00040
C COMMON /PATHL/ X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20) LIS00050
C COMMON /LEVEL/ ISLTUP,ISLTDN,IHORIZ,IVERT,TESSL,TESARG,TESVT LIS00060
C COMMON /BASTOP/ ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10) LIS00070
C COMMON /BASTH/ ZLBASE(10),ZLTHICK(10),ZCBASE(10),ZCTHICK(10) LIS00080
C +,RADICL(10)
C COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10) LIS00090
C COMMON /IOUNIT/ IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUL LIS00100
C DATA ALPH/2HST,2HAS,2HNS,2HSC,2HCH,2HCC/
C IF(IWV.EQ.1)WVL=0.55 LIS00110
C IF(IWV.EQ.2)WVL=3.80 LIS00120
C IF(IWV.EQ.3)WVL=10.60 LIS00130
C IF(IRUN.LT.2) WRITE(IOOUT,150)WVL LIS00140
C IF(IRUN.GT.1) WRITE(IOOUT,160)WVL LIS00150
150 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS //) LIS00160
160 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS //) LIS00170
IF(IHORIZ.EQ.1)WRITE(IOOUT,200) LIS00180
IF(IVERT.EQ.1)WRITE(IOOUT,210) LIS00190
IF(ISLTUP.EQ.1)WRITE(IOOUT,220) LIS00200
IF(ISLTDN.EQ.1)WRITE(IOOUT,230) LIS00210
200 FORMAT(1H0,40X,28HLINE-OF-SIGHT IS HORIZONTAL //) LIS00220
210 FORMAT(1H0,40X,26HLINE-OF-SIGHT IS VERTICAL //) LIS00230
220 FORMAT(1H0,40X,28HLINE-OF-SIGHT SLANTS UPWARD //) LIS00240
230 FORMAT(1H0,40X,30HLINE-OF-SIGHT SLANTS DOWNWARD //) LIS00250
PTHLEN=SQRT((XS-XT)**2+(YS-YT)**2+(ZS-ZT)**2) LIS00260
WRITE(IOOUT,300)PTHLEN LIS00270
300 FORMAT(1H0,40X,29HTOTAL LINE-OF-SIGHT LENGTH = ,F7.3,4H KM ) LIS00280
ATTLEN=0.0 LIS00290
IF(TAUTOT.EQ.0.0)GO TO 780 LIS00300
DO 310 N=1,NN LIS00310
310 ATTLEN=ATTLEN+ATTL(N) LIS00320
WRITE(IOOUT,400)ATTLEN LIS00330
400 FORMAT(1H0,40X,50HTOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD LIS00340
+ = ,F7.3,4H KM ) LIS00350
WRITE(IOOUT,500)TAUTOT LIS00360
500 FORMAT(1H0,40X,22HTOTAL OPTICAL DEPTH = ,F7.2//) LIS00370
WRITE(IOOUT,600)CTRANS LIS00380
600 FORMAT(1H0,40X,36HTRANSMITTANCE ALONG LINE-OF-SIGHT = ,E11.5,//) LIS00390
WRITE(IOOUT,605) LIS00400
605 FORMAT(1H0,40X,50HSEEKER COORDINATES (KM) TARGET COORDINATES (KMLIS00410
+ ) LIS00420
WRITE(IOOUT,606) LIS00430
606 FORMAT(1H ,40X,23HXSEEKER YSEEKER ZSEEKER,3X,23HXTARGET YTARGET ZT LIS00440
+ ARGET) LIS00450
WRITE(IOOUT,607) LIS00460
607 FORMAT(1H ,40X,3(8H----- ),2X,3(8H----- )) LIS00470
WRITE(IOOUT,608)XS,YS,ZS,XT,YT,ZT LIS00480
608 FORMAT(1H0,40X,3(F6.3,2X),2X,3(F6.3,2X)) LIS00490
WRITE(IOOUT,613) LIS00500
613 FORMAT(1H //) LIS00510
WRITE(IOOUT,615) LIS00520
615 FORMAT(1H0,40X,62HCLOUD TYPE LINE-OF-SIGHT INTERSECTION COORD LIS00530
+ INATES (KM) ) LIS00540
WRITE(IOOUT,620) LIS00550
620 FORMAT(1H ,40X,10H/ID NUMBER,4X,47HXUPPER YUPPER ZUPPER XLOWER LIS00560
+ YLOWER ZLOWER) LIS00570
WRITE(IOOUT,625) LIS00580
625 FORMAT(1H ,40X,10H----- ,4X,3(8H----- ),1X,3(8H----- )) LIS00590
IF(NLAYER.EQ.0)GO TO 685 LIS00600
DO 670 N=1,NLAYER LIS00610
IND=ILCTYP(N) LIS00620
IF(NLINT(N).EQ.0)GO TO 670 LIS00630
WRITE(IOOUT,680)ALPH(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N) LIS00640
670 CONTINUE LIS00650
680 FORMAT(1H0,42X,A2,1H/,I2,6X,3(F7.3,1X),1X,3(F7.3,1X)) LIS00660
LIS00670
LIS00680
LIS00690
LIS00700

```

```

685 NLO=NLAYER+1           LIS00710
    IF(NLAYER.EQ.NN)GO TO 699
    LL=0                   LIS00720
    DO 690 N=NLO,NN        LIS00730
    LL=LL+1                LIS00740
    IND=ICCTYP(LL)         LIS00750
    IF(NCINT(LL).EQ.0)GO TO 690
    WRITE(I0OUT,680)ALPH(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N) LIS00760
690 CONTINUE                LIS00770
    WRITE(I0OUT,613)          LIS00780
    WRITE(I0OUT,691)          LIS00790
691 FORMAT(1H0,40X,59H CLOUD TYPE   CUMULUS CENTER POSITIONS (FOR CC ORL LIS00820
  + CH TYPES) )            LIS00830
    WRITE(I0OUT,692)          LIS00840
692 FORMAT(1H,40X,10H / ID NUMBER,6X,26HX CLOUD (KM)      Y CLOUD (KM)) LIS00850
    WRITE(I0OUT,693)          LIS00860
693 FORMAT(1H,40X,10H-----,6X,26H-----)               LIS00870
    LL=0                   LIS00880
    DO 695 N=NLO,NN        LIS00890
    LL=LL+1                LIS00900
    IND=ICCTYP(LL)         LIS00910
695 WRITE(I0OUT,696)ALPH(IND),N,XCLOUD(LL),YCLOUD(LL) LIS00920
696 FORMAT(1H0,42X,A2,1H/,I2,10X,F7.3,8X,F7.3)          LIS00930
699 CONTINUE                LIS00940
    WRITE(I0OUT,613)          LIS00950
    DO 610 N=1,NN           LIS00960
610 T(N)=EXP(-TAU(N))       LIS00970
    IF(NCCLDS.GT.0) WRITE(I0OUT,700)
    IF(NCCLDS.LT.1) WRITE(I0OUT,705)
700 FORMAT(1H1,20X,11H CLOUD TYPE ,2X,15H HEIGHT OF BASE ,2X,
  +10H THICKNESS ,2X,16H RADIUS OF CLOUD ,2X,14H OPTICAL DEPTH ,2X,
  +14H TRANSMITTANCE )     LIS01000
705 FORMAT(1H0,20X,11H CLOUD TYPE ,2X,15H HEIGHT OF BASE ,2X,
  +10H THICKNESS ,2X,16H RADIUS OF CLOUD ,2X,14H OPTICAL DEPTH ,2X,
  +14H TRANSMITTANCE )     LIS01010
    WRITE(I0OUT,710)          LIS01020
710 FORMAT(1H ,20X,10H / ID NUMBER,9X,4H(KM),9X,4H(KM),12X,4H(KM),9X,
  +11HALONG L-O-S,5X,11HALONG L-O-S)                      LIS01030
    WRITE(I0OUT,720)          LIS01040
720 FORMAT(1H ,20X,10H-----,3X,14H-----,3X,9H-----,
  +3X,15H-----,3X,13H-----,3X,13H-----)               LIS01050
    IF(NLAYER.EQ.0)GO TO 735
    BLANK=0,0             LIS01060
    DO 730 N=1,NLAYER      LIS01070
    IND=ILCTYP(N)          LIS01080
730 WRITE(I0OUT,760)ALPH(IND),N,ZLBASE(N),ZLTHIC(N),BLANK,TAU(N),T(N) LIS01090
735 NLO=NLAYER+1          LIS01100
740 IF(NN.EQ.NLAYER)GO TO 800
    LL=0                   LIS01110
    DO 750 N=NLO,NN        LIS01120
    LL=LL+1                LIS01130
    IND=ICCTYP(LL)         LIS01140
750 WRITE(I0OUT,760)ALPH(IND),N,ZCBASE(LL),ZCTHIC(LL),RADIDL(LL),
  +TAU(N),T(N)           LIS01150
760 FORMAT(1H0,22X,A2,1H/,I2,10X,F7.3,7X,F7.3,8X,F7.3,10X,F7.2,7X,
  +E11,5)                LIS01160
    GO TO 800              LIS01170
780 WRITE(I0OUT,790)          LIS01180
790 FORMAT(1H0,20X,59H NO CLOUD OBSCURATION : L-O-S DOES NOT INTERSECT LIS01190
  +ANY CLOUDS //)          LIS01200
800 RETURN                 LIS01210
END                         LIS01220

```

```

SUBROUTINE DEFSET(ISTEP,NMAX) DEF00010
C***** THIS ROUTINE RESETS THE CLOUD BASE HEIGHT, THICKNESS, DEF00020
C***** AND RADIUS VALUES TO THE NEAREST REALISTIC BOUNDARIES DEF00030
C***** IF THEY DO NOT LIE CLOSE TO THE RANGES SPECIFIED IN DEF00040
C***** R. D. H. LOW'S PAPER. DEF00050
CDEF00060
CDEF00070
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10) DEF00080
COMMON /BASTH/ZLBASE(10),ZLTHICK(10),ZCBASE(10),ZCTHICK(10) DEF00090
+,RADICL(10) DEF00100
IF(ISTEP.NE.1)GO TO 10 DEF00110
DO 5 N=1,NMAX DEF00120
ITYPE=ILCTYP(N) DEF00130
IF(ITYPE.EQ.2)GO TO 3 DEF00140
1 IF(ZLBASE(N).LT.0.1)ZLBASE(N)=0.1 DEF00150
IF(ZLBASE(N).GT.1.5)ZLBASE(N)=1.5 DEF00160
IF(ITYPE.EQ.3)GO TO 4 DEF00170
2 IF(ZLTHICK(N).LT.0.2)ZLTHICK(N)=0.2 DEF00180
IF(ZLTHICK(N).GT.1.0)ZLTHICK(N)=1.0 DEF00190
GO TO 5 DEF00200
3 IF(ZLBASE(N).LT.2.0)ZLBASE(N)=2.0 DEF00210
IF(ZLBASE(N).GT.5.0)ZLBASE(N)=5.0 DEF00220
4 IF(ZLTHICK(N).LT.1.0)ZLTHICK(N)=1.0 DEF00230
IF(ZLTHICK(N).GT.4.0)ZLTHICK(N)=4.0 DEF00240
5 CONTINUE DEF00250
10 IF(ISTEP.NE.2)GO TO 20 DEF00260
DO 15 N=1,NMAX DEF00270
IF(ZCBASE(N).LT.0.8)ZCBASE(N)=0.8 DEF00280
IF(ZCBASE(N).GT.1.5)ZCBASE(N)=1.5 DEF00290
IF(ZCTHICK(N).LT.0.2)ZCTHICK(N)=0.2 DEF00300
IF(ZCTHICK(N).GT.5.0)ZCTHICK(N)=5.0 DEF00310
IF(RADICL(N).LT.0.05>RADICL(N)=0.05 DEF00320
IF(RADICL(N).GT.0.6>RADICL(N)=0.6 DEF00330
15 CONTINUE DEF00340
20 RETURN DEF00350
END DEF00360

```

```

SUBROUTINE CLREAD(NLAYER,NCCLDS,IERR)
DIMENSION IALPHA(18),DATELT(5) CLR00010
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20) CLR00020
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20) CLR00030
COMMON /GEOMET/PTS(15),IGEOSW CLR00040
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10) CLR00050
COMMON /IOUNIT/IJOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU CLR00070
COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10) CLR00080
+ RAPIDCL(10) CLR00090
COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10) CLR00100
DATA IALPHA/2HCL,2HST,2HCL,2HAS,2HCL,2HNS,2HCL,2HSC,2HCL,2HCH,
+ 2HCL,2HCC,2HSE,2HEK,2HTA,2HRG,2HGO,2H/ CLR00110
ISFLAG=0 CLR00120
ITFLAG=0 CLR00130
IF(IGEOSW.NE.1)GO TO 60 CLR00140
XS=PTS(4) CLR00150
YS=PTS(5) CLR00160
ZS=PTS(6) CLR00170
XT=PTS(1) CLR00180
YT=PTS(2) CLR00190
ZT=PTS(3) CLR00200
60 CONTINUE CLR00220
DO 900 N=1,23 CLR00230
READ(IJOIN,100)IALPH,IALP,(DATELT(I),I=1,5) CLR00240
100 FORMAT(2A2,1X,5E10.5,1X) CLR00250
INDEX=10 CLR00260
DO 200 K=1,17,2 CLR00270
IF((IALPH.EQ.IALPH(K)).AND.(IALP.EQ.IALPH(K+1))) GO TO 50 CLR00280
GO TO 200
50 REALK=FLOAT(K)
ROT=REALK/2.0
KJ=IFIX(ROT)+1
INDEX=KJ
GO TO 300
200 CONTINUE
300 IF(INDEX.LT.7)GO TO <350,350,350,350,400,400>,INDEX
INM6=INDEX-6
GO TO <500,600,999,997>,INM6
350 NLAYER=NLAYER+1
IF(NLAYER.GT.10) GO TO 993
ILCTYP(NLAYER)=INDEX
ZLBASE(NLAYER)=DATELT(1)
ZLTHIC(NLAYER)=DATELT(2)
GO TO 900
400 NCCLDS=NCCLDS+1
IF(NCCLDS.GT.10) GO TO 995
ICCTYP(NCCLDS)=INDEX
ZCBASE(NCCLDS)=DATELT(1)
ZCTHIC(NCCLDS)=DATELT(2)
RAPIDCL(NCCLDS)=DATELT(3)
XCLOUD(NCCLDS)=DATELT(4)
YCLOUD(NCCLDS)=DATELT(5)
GO TO 900
500 IF(ISFLAG.EQ.1) GO TO 997
XS=DATELT(1)
YS=DATELT(2)
ZS=DATELT(3)
ISFLAG=1
GO TO 900
600 IF(ITFLAG.GT.1) GO TO 997
XT=DATELT(1)
YT=DATELT(2)
ZT=DATELT(3)
ITFLAG=1
900 CONTINUE
GO TO 999
993 IERR=1
WRITE(IOOUT,994)
994 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF STRATIFORM CLOUDS ICLR00650
+INPUT EXCEEDS THE LIMIT OF 10 //) CLR00660

```

```
GO TO 999                                CLR00670
995 IERR=1                                CLR00680
      WRITE( IOOUT, 996)                      CLR00690
996 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF CUMULIFORM CLOUDS ICLR00700
      +INPUT EXCEEDS THE LIMIT OF 10 //)      CLR00710
      GO TO 999                                CLR00720
997 IERR=1                                CLR00730
      WRITE( IOOUT, 998)                      CLR00740
998 FORMAT(1H0,20X,64H***CLREAD ERROR*** IMPROPER INPUT FORMAT OR ABSECLR00750
      +NT GO    SENTINEL //)                  CLR00760
999 RETURN                                CLR00770
      END                                     CLR00780
```

```

SUBROUTINE SCREEN(IERR,ICLMAT) SCR00010
THIS MODULE 1) COMPUTES TRANSMITTANCE REQUIRED TO REDUCE THE SCR00020
PROBABILITY OF STATIC TARGET DETECTION BELOW A SCR00030
GIVEN LEVEL FOR CERTAIN TYPES OF IMAGERS. (ITAM) SCR00040
SCR00050
SCR00060
SCR00070
SCR00080
SCR00090
SCR00100
SCR00110
SCR00120
SCR00130
SCR00140
SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

- AND/OR -
2) COMPUTES HC AND WP 105 MM/155 MM SMOKE MUNITIONS SCR00090
REQUIRED TO PRODUCE A SMOKE SCREEN OF USER-DEFINED SCR00100
LENGTH AND DURATION FOR VISIBLE, NEAR, MID AND FAR SCR00110
IR WAVELENGTHS. SCR00120

SUBROUTINES CALLED BY SCREEN ARE - CWIC AND ITAM. SCR00130
ALL OUTPUT FROM SCREEN IS TABULAR. SCR00140
SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

THE PRESENT VERSION OF CWIC USES THE XSCALE MODULE TO OPTIONALLY SCR00090
CORRECT FOR EXTINCTION DUE TO FOG, RAIN AND/OR SNOW AT IR SCR00100
WAVELENGTHS BASED ON VISIBILITY IN THE .5 MICROMETER REGION. SCR00110
CWIC CAN ALSO COMPUTE PASQUILL CATEGORY FROM FUNDAMENTAL MET INPUTS SCR00120
IF THE USER CHOOSES NOT TO PROVIDE THE CATEGORY DIRECTLY, SCR00130
CLIMATOLOGICAL VALUES FROM THE CLIMAT MODULE CAN OPTIONALLY BE SCR00140
USED AS "TYPICAL" MET INPUTS. SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

THE ITAM MODULE CAN BE USED IN A LOOPING MODE THROUGH MULTIPLE SETS SCR00090
OF INPUT RECORDS TO GENERATE TABLES. THE LAST VALUE OF TRANS- SCR00100
MITTANCE COMPUTED IS THAT VALUE WHICH CAN (OPTIONALLY) BE PASSED SCR00110
TO CWIC AS THE THRESHOLD LEVEL FOR TOTAL PATH TRANSMITTANCE. SCR00120
SCR00130
SCR00140
SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

TWO RECORDS MUST BE PROVIDED TO SCREEN: SCR00090
ONE CARD MUST BE INPUT TO SCREEN TO SELECT OPTIONS: (3(1X,I1)) SCR00100
COL 2 ICITAM =1 CALL ITAM, OR 0 (NO CALL). SCR00110
COL 4 ICCWIC =1 CALL CWIC, OR 0 (NO CALL). SCR00120
COL 6 ICCLIM =1 USE CLIMAT FOR MET INPUTS, OR 0 USE USER VALUES. SCR00130
SCR00140
SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

IF CHOSEN, ALL INPUT RECORDS TO ITAM ARE READ FOLLOWING THE ABOVE SCR00090
RECORD. (SEE INPUT DESCRIPTION IN ITAM) SCR00100
AND, IF CHOSEN, INPUT RECORDS TO CWIC ARE THEN READ. (SEE INPUT SCR00110
DESCRIPTION IN CWIC) SCR00120
SCR00130
SCR00140
SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

FINALLY, ONE RECORD IS READ BY SCREEN WITH THE WORD "END" IN COLUMNS SCR00090
1-3. THIS RETURNS CONTROL TO THE EOSAEL EXECUTIVE MODULE. SCR00100
SCR00110
SCR00120
SCR00130
SCR00140
SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00195
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00295
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

COMMON /IOUNIT/I0IN, I0OUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTS
10 READ (I0IN,10) ICITAM, ICCWIC, ICCLIM
FORMAT(1X,I1,1X,I1,1X,I1)
IF (ICITAM.NE.0) CALL ITAM(IERR,TFNL)
IF (IERR.EQ.1) GOTO 15
IF (ICCLIM.EQ.0) GOTO 20
IF (ICLMAT.EQ.1) GOTO 20
IERR=1
WRITE (I0OUT,30)
30 FORMAT(1X,62H*** IN SCREEN ROUTINE, MET SOURCE SPECIFIED AS CLIMAT
*OLOGICAL,/5X,39HBUT CLIMAT ROUTINE HAD NOT BEEN CALLED.)
RETURN
15 WRITE (I0OUT,40)
TFNL=1
40 FORMAT(1X,21HIERR FLAG SET IN ITAM)
IF (ICCWIC.NE.0) CALL CWIC (IERR,ICITAM,ICCLIM,TFNL)
RETURN
20 END
END

```

SUBROUTINE CWIC (IERR, ICITAM, ICCLIM, TFNL) CWC00010
 CWIC COMPUTES THE REQUIRED SMOKE MUNITIONS EXPENDITURE (NUMBER, CWC00020
 RATE OF FIRE, PLACEMENT) TO PRODUCE A SCREEN OF DEFINED CWC00030
 LENGTH AND DURATION USING HC OR WP 105MM OR 155MM SMOKE CWC00040
 MUNITIONS. WAVELENGTHS ARE FOR RANGES OF VISIBLE, NEAR, CWC00050
 MID AND FAR IR. CWC00060
 CWC00070
 CWC00080
 CWC00090
 CWC00100
 CWC00110
 CWC00120
 CWC00130
 CWC00140
 CWC00150
 CWC00160
 CWC00170
 CWC00180
 CWC00190
 CWC00200
 CWC00210
 CWC00220
 CWC00230
 CWC00240
 CWC00250
 CWC00260
 CWC00270
 CWC00280
 CWC00290
 CWC00300
 CWC00310
 CWC00320
 CWC00330
 CWC00340
 CWC00350
 CWC00360
 CWC00370
 CWC00380
 CWC00390
 CWC00400
 CWC00410
 CWC00420
 CWC00430
 CWC00440
 CWC00450
 CWC00460
 CWC00470
 CWC00480
 CWC00490
 CWC00500
 CWC00510
 CWC00520
 CWC00530
 CWC00540
 CWC00550
 CWC00560
 CWC00570
 CWC00580
 CWC00590
 CWC00600
 CWC00610
 CWC00620
 CWC00630
 CWC00640
 CWC00650
 CWC00660
 CWC00670
 CWC00680
 CWC00690
 CWC00700

ON INPUT, IF ICITAM IS NON-ZERO, ALL COMPUTATIONS WILL USE CWC00080
 THE INPUT TFNL AS THE TOTAL SCREEN TRANSMISSION THRESHOLD. IF CWC00090
 ICITAM IS 0, THEN BUILT-IN TOTAL THRESHOLDS OF .05 ARE USED CWC00100
 FOR ALL WAVELENGTH REGIONS. CWC00110
 CWC00120
 CWC00130
 CWC00140
 CWC00150
 CWC00160
 CWC00170
 CWC00180
 CWC00190
 CWC00200
 CWC00210
 CWC00220
 CWC00230
 CWC00240
 CWC00250
 CWC00260
 CWC00270
 CWC00280
 CWC00290
 CWC00300
 CWC00310
 CWC00320
 CWC00330
 CWC00340
 CWC00350
 CWC00360
 CWC00370
 CWC00380
 CWC00390
 CWC00400
 CWC00410
 CWC00420
 CWC00430
 CWC00440
 CWC00450
 CWC00460
 CWC00470
 CWC00480
 CWC00490
 CWC00500
 CWC00510
 CWC00520
 CWC00530
 CWC00540
 CWC00550
 CWC00560
 CWC00570
 CWC00580
 CWC00590
 CWC00600
 CWC00610
 CWC00620
 CWC00630
 CWC00640
 CWC00650
 CWC00660
 CWC00670
 CWC00680
 CWC00690
 CWC00700

INPUTS TO CWIC ARE ON STANDARDIZED RECORDS CONTAINING KEY-WORDS CWC00140
 IN COLUMNS 1-4 AND REAL (IE DECIMAL) VALUES FOR ALL INPUTS, CWC00150
 PLACED IN FIELDS 11-20, 21-30, ..., 71-80. KEY-WORD TYPES CWC00160
 ARE SCRN FOR SCREEN, OBSERVER/TARGET LOS AND ADVERSE CWC00170
 WEATHER CORRECTIONS.
 METR FOR METEOROLOGICAL CONDITIONS (MAY BE OMITTED CWC00180
 IF ICCLIM IS NON-ZERO, IN WHICH CASE THE CWC00190
 CLIMATOLOGICAL VALUES FROM CLIMAT ARE USED.
 PASQ FOR (OPTIONAL) MET PARAMETERS REQUIRED TO COMPUTE CWC00200
 THE PASQUILL STABILITY CATEGORY, NOT REQUIRED CWC00210
 IF PASQUILL CATEGORY ITSELF IS INPUT.
 DONE WHICH RETURNS EXECUTIVE CONTROL BACK TO THE SCREEN CWC00220
 MODULE.
 THE ORDER OF THE INPUT RECORDS IS IMMATERIAL, EXCEPT THAT THE (DONE) CWC00230
 CARD MUST BE LAST.
 INPUTS: (ALL VALUES REAL) FORMAT (2A2,6X,7F10.3)
 KEYWORD COLS. VARIABLE DESCRIPTION

KEYWORD	COLS.	VARIABLE	DESCRIPTION
SCRN	1-4		SCREEN/LOS DEFINITION (REQUIRED)
	11-20	TIME	- SCREEN DURATION (MINUTES)
	21-30	X0	- SCREEN LENGTH (METERS)
	31-40	H3	- SLANT RANGE OBS-TARGET (KM)
	41-50	AST	- ELEVATION ANGLE OF TARGET FROM OBSERVER WRT HORIZONTAL (DEG.)
	51-60	DLS	- COMPASS DIRECTION (CLOCKWISE WRT NORTH) OF LINE-OF-SIGHT (DEG.)
	61-70	ARE	- TERRAIN ROUGHNESS ELEMENT (CM)
	71-80	FOG	- ADVERSE WEATHER/HAZE SELECTION 0. = NO ADVERSE WEATHER 1. = ONLY CORRECT VISIBLE WAVE- LENGTHS FOR INPUT VISIBILITY. 2. = CORRECT FOR FOG/HAZE FOR MARITIME ARCTIC AIR MASS 3. = CORRECT FOR FOG/HAZE FOR MARITIME POLAR AIR MASS 4. = CORRECT FOR FOG/HAZE FOR CONTINENTAL POLAR AIR MASS 5. = CORRECT FOR RAIN. 6. = CORRECT FOR SNOW.
METR	1-4		MET INPUTS (NOT REQUIRED IF ICCLIM NON-ZERO FROM SCREEN MODULE.)
	11-20	S3	- WINDSPEED (METERS/SEC)
	21-30	D0	- WIND DIRECTION (DEG) CLOCKWISE WRT NORTH, USUAL MET CONVENTION
	31-40	PCAT	- PASQUILL CATEGORY, IF INPUT AS 0, SEE PASQ RECORD BELOW. OTHERWISE, 1.=A, 2.=B, 3.=C, 4.=D, 5.=E, 6.=F
	41-50	VS	- VISIBILITY (KM) NOT REQUIRED IF FOG = 0. IS SPECIFIED.
	51-60	R0	- RELATIVE HUMIDITY (PERCENT). IF 0, THEN DEW POINT AND TEMPERATURE ARE REQUIRED BELOW TO COMPUTE R0.
	61-70	T0	- AIR TEMP. (DEG C) REQUIRED IF R0 NO GIVEN.

	71-80	T1	- DEW POINT TEMP. <DEG C> REQUIRED IF CWIC00710 R0 NOT GIVEN.	CWC00720 CWC00730
PASQ	1-4		PASQUILL CATEGORY DETERMINATION (REQUIWC00740 RED ONLY IF PCAT ABOVE IS 0.)	CWC00750
	11-20	SLAT	- LATITUDE OF SITE <DEG.>, POSITIVE ICWC00760 NORTH LATITUDE	CWC00770
	21-30	SLONG	- LONGITUDE OF SITE <DEG.>, POSITIVE CWC00780 EAST LONGITUDE	CWC00790
	31-40	SJDATE	- JULIAN DATE, <DECIMAL DAYS>	CWC00800
	41-50	SZHOUR	- GMT TIME OF DAY <DECIMAL HOURS AND FRACTIONAL HOURS>	CWC00810 CWC00820
	51-60	C0	- CEILING CLOUD HEIGHT <METERS>	CWC00830
	61-70	C1	- CLOUD COVER <PERCENT>	CWC00840
DONE	1-4		END OF CWIC EXECUTION (REQUIRED)	CWC00850 CWC00860
ALL OUTPUT FROM CWIC IS TABULAR. SUBROUTINES CALLED ARE -				CWC00870 CWC00880 CWC00890
CWIC1, CWIC3, CWIC4, JPASCT, XSCALE.				CWC00900
DIMENSION CS(4,2), TRSH(4), WAVE(4), ISLANT(4), IR(8), *H(2,2), U(2,2), LNGTH(16), ISMOKE(2), IGUN(4), JANS(4)				CWC00910 CWC00920
COMMON /IOUNIT/IOUT, IOIN, IOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUWC0093				CWC00940
COMMON /GEOMET/PTS(15), IGEOISW				CWC00950
COMMON /CLYMAT/ TEMP, PRESS, RH, AH, DP, VIS, CLDAMT, CLDHYT, FOGPRB, *WNDVEL, WNDDIR, IPASCT				CWC00960
COMMON /M06/ P(4,2,2), Q1(4,2,2), Q2(4,2,2), Y(4,2,2), Z(4,2,2), * C(4,2), R(4,2), V(2), RV(7), T(4), ARE, C0, C1, C2, D0, D2,				CWC00970 CWC00980
* DLS, R0, R2, S0, S3, SJDATE, SLAT, SLONG, SZHOUR, TIME, * TR(4), X0, Y1, IT(2), IP0, IDUMMY				CWC00990
COMMON /CONST/ PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK				CWC01000
EXTERNAL JPASCT				CWC01020
DATA IR/2HSC, 2HBN, 2HME, 2HTR, 2HPA, 2HSQ, 2HD0, 2HNE/ DATA H /18.7, 77.1, 1737.3, 7076.2/				CWC01030 CWC01040
DATA U /5.4, 7.9, 1.8, 2.6/				CWC01050
DATA LNGTH /2HVI, 2HSI, 2HBL, 2HE/, 2HNE, 2HAR, 2H I, 2HR/, 2HMI, *2HD, 2HIR, 2H, 2HFA, 2HR, 2HIR, 2H/				CWC01060 CWC01070
DATA JANS /2HN0, 2H, 2HYE, 2HS/				CWC01080
DATA ISMOKE /2HHC, 2HWP/				CWC01090
DATA IGUN /2H10, 2H5, 2H15, 2H5/				CWC01100
C *** INITIALIZE VALUES FOR DEFAULTS.				CWC01110
EXTINCTION COEF HC <VIS., NEAR, MID, FAR>, WP <VIS., NEAR, MID, FAR IR>				CWC01120
DATA CS /3.3, 1.5, 0.53, 0.13, 2.46, 2.0, .25, .32/				CWC01130
BUILT IN TRANSMISSION THRESHOLDS, WAVELENGTHS, XSCALE SLANTS				CWC01140
DATA TRSH /.05, .05, .05, .05/				CWC01150
DATA WAVE /.55, 1.06, 3.5, 10.6/, ISLANT/0, 1, 1, 1/				CWC01160
ACOS(ARG)=ATAN2(SQRT(1.-ARG**2), ARG)				
TIME=10.				CWC01170
X0=1000.				CWC01180
H3=3.				CWC01190
AST=0.				CWC01200
DLS=90.				CWC01210
ARE=1.				CWC01220
FOG=0.				CWC01230
S3=4.				CWC01240
DU=0.				CWC01250
PCAT=3.				CWC01260
R0=50.				CWC01270
TG=24.				CWC01280
T1=12.94				CWC01290
VS=10.				CWC01300
C READ DATA CARDS				CWC01310
ICOU=0				CWC01320
10	READ (IOIN, 900) IT(1), IT(2), (RV(I), I=1, 7)			CWC01330
	ICOU=ICOU+1			
	IF (IT(1).EQ.IR(1).AND.IT(2).EQ.IR(2)) GOTO 20			CWC01340
	IF (IT(1).EQ.IR(3).AND.IT(2).EQ.IR(4)) GOTO 40			CWC01350
	IF (IT(1).EQ.IR(5).AND.IT(2).EQ.IR(6)) GOTO 50			CWC01360
	IF (IT(1).EQ.IR(7).AND.IT(2).EQ.IR(8)) GOTO 60			CWC01370
	WRITE (IOUT, 901) IT(1), IT(2), (RV(I), I=1, 7)			CWC01380
				CWC01390

```

900  FORMAT(2A2,6X,7F10.3)          CWC01400
901  FORMAT(1X,51HIN CWIC, THE FOLLOWING CARD DOES NOT CONFORM TO PRO, CWC01410
*15HPER CONVENTIONS/1X,2A2,6X,7F10.3)          CWC01420
IF (ICOU.LE.4) GOTO 10          CWC01430
WRITE (IOOUT,902)          CWC01440
902  FORMAT(1X,39HINVALID INPUTS TO CWIC. IERR=1 RETURNED)          CWC01450
IERR=1          CWC01460
RETURN          CWC01470
20   TIME=RV(1)          CWC01480
X0=RV(2)          CWC01490
H3=RV(3)          CWC01500
AST=RV(4)          CWC01510
DLS=RV(5)          CWC01520
ARE=RV(6)          CWC01530
FOG=RV(7)          CWC01540
IFK ICEOSH,NE,1) GO TO 22          CWC01550
DELX=PTS(1)-PTS(4)
DELY=PTS(2)-PTS(5)
DELZ=PTS(3)-PTS(6)
H3=SQRT(DELX**2+DELY**2+DELZ**2)
HDIS=SQRT(DELX**2+DELY**2)
RTDCON=57.29577951
AST=RTDCON*ACOS(HDIS/H3)
IF(HDIS.GT.1.E-20) DLS=RTDCON*ACOS(DELY/HDIS)
IF(DELX.LT.0.) DLS=360.-DLS
22   CONTINUE          CWC01640
GOTO 10          CWC01650
40   S3=RV(1)          CWC01660
D0=RV(2)          CWC01670
PCAT=RV(3)          CWC01680
VS=RV(4)          CWC01690
R0=RV(5)          CWC01700
T0=RV(6)          CWC01710
T1=RV(7)          CWC01720
GOTO 10          CWC01730
50   SLAT=RV(1)          CWC01740
SLONG=RV(2)          CWC01750
SJDATE=RV(3)          CWC01760
SZHOUR=RV(4)          CWC01770
C0=RV(5)          CWC01780
C1=RV(6)          CWC01790
GOTO 10          CWC01800
C*** BEGIN COMPUTATIONS. -- FIRST MET VALUES.
60   IF (ICCLIM.EQ.0) GOTO 70          CWC01810
C*** USE CLIMAT PASSED VALUES:
T0=TEMP
T1=DP
R0=RH
PCAT=FLOAT(IPASCT)
VS=YIS
D0=WNDDIR
S3=WNDVEL
C*** PROVIDE WINDSPEED IN KNOTS AS S0:
70   S0=S3/.515          CWC01820
IF (S0.LE.1.) S3=.515          CWC01830
IF (S0.LE.1.) S0=1          CWC01840
CWC01850
CWC01860
CWC01870
CWC01880
CWC01890
CWC01900
CWC01910
CWC01920
CWC01930
CWC01940
CWC01950
CWC01960
CWC01970
CWC01980
CWC01990
CWC02000
CWC02010
CWC02020
CWC02030
CWC02040
CWC02050
CWC02060
CWC02070
CWC02080
CWC02090
C*** NOW CHECK RH AND COMPUTE IF NECESSARY.
IRNOT=0
IF (R0.GT.0.) GOTO 80
IF (T0.GT.0.) GO TO 76
A0=9.5
B0=265.5
IF (T0.LE.0.) GO TO 78
CONTINUE
A0=7.5
B0=237.3
CONTINUE
IF (T1.GE.0.) GO TO 79
A1=9.5
B1=265.5

```

```

IF(T1.LE.0.) GO TO 75
CONTINUE
A1=7.5
B1=237.3
CONTINUE
E0=6.11*10.***((A0*T0)/(B0+T0))
E1=6.11*10.***((A1*T1)/(B1+T1))
R0=(E1/E0)*100.
IPNOT=1
C*** NOW CHECK PASQUILL CATEGORY..
80   IPO=IFIX(PLAT+.001)
     IF (IPO.GT.6) IPO=6
     IPNOT=0
     IF (IPO.GT.0) GOTO 90
     CALL CWIC1 (IPO,C0,C1,SLAT,SLONG,SJDATE,SZHOUR,S0)
     IPNOT=1
C*** NEXT COMPUTE CL VALUE FOR THRESHOLD TRANSMITTANCE, CORRECTED FOR
FOG,RAIN,.
C*** VISIBILITY EXTINCTION (KM-1). NOTE THAT IF VS=0., THEN SET TO
CLEAR DAY AND COMPUTATIONS CONTINUE.
90   EX55=0.
     IF (VS.GT.0.) EX55=3.912/VS
C*** TRANSMISSION THRESHOLDS..
DO 92 I=1,4
     T(I)=TRSH(I)
     IF (ICITAM.NE.0) T(I)=TFNL
     TRC(I)=T(I)
92   CONTINUE
C*** CORRECTIONS FOR WEATHER
     IFOG=-1+IFIX(FOG+.0001)
     IF (IFOG.LE.-1) GOTO 100
C*** CORRECT VISIBLE FOR VISIBILITY (H3 SLANT RNG, AST ELEV. ANG.)
XSTRN=0.
     EPTH=EX55*H3
     IF (EPHT.LT.12.) XSTRN=EXP(-EPHT)
C SET VSET FOR NO CARD I/O IN XSCALE
     VSET=89.
     IF((AST.GT.0.1.OR. AST.LT.-0.1).AND.IFOG.LT.5)
*CALL XSCALE(WAVE(I),VSET,EX55,XSTRN,IERR,ISLANT(I),IFOG,H3,AST)
     IF (XSTRN.LE.0.) XSTRN=.0001
     IF (XSTRN.GT.1.) XSTRN=1.
     T(I)=T(I)/XSTRN
     IF (IFOG.EQ.0) GOTO 100
C CORRECT NON-VISIBLE FOR SEEABILITY.
94   CONTINUE
DO 96 I=2,4
     ISLNT=ISLANT(I)
     IF (AST.EQ.0.) ISLNT=0
     IF (IFOG.GE.5) ISLNT=0
     CALL XSCALE(WAVE(I),VSET,EX55,XSTRN,IERR,ISLNT,IFOG,H3,AST)
     IF (XSTRN.LE.0.) XSTRN=.0001
     IF (XSTRN.GT.1.) XSTRN=1.
     T(I)=T(I)/XSTRN
96   CONTINUE
C*** COMPUTE CL FOR THRESHOLD TRANSMISSION REQUIRED OF SMOKE.
100  DO 108 I=1,4
     IF (T(I).GT.1.) T(I)=1.
     IF (T(I).LE.0.) T(I)=.00001
     DO 105 J=1,2
         C(I,J)= ALOG(T(I))/(-CS(I,J))
105  CONTINUE
108  CONTINUE
C*** ATMOSPHERIC DIFFUSION
     CALL CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U,V,Y1)
C*** MUNITIONS EXPENDITURES
     CALL CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0)
C** END MAIN CWIC COMPUTATIONS, FINAL OUTPUT.
     IF(ICITAM.GT.0) WRITE(I00UT,10000)
     WRITE(I00UT,10200)
     WRITE(I00UT,10800)

```

```

        WRITE (IOOUT,10140)                                     CWC02810
        WRITE (IOOUT,10300)                                     CWC02820
        WRITE (IOOUT,10400)                                     CWC02830
        WRITE (IOOUT,10200)                                     CWC02840
        WRITE (IOOUT,14000) H3                                CWC02850
        WRITE (IOOUT,14100) AST                               CWC02860
        WRITE (IOOUT,14200) DLS                               CWC02870
        WRITE (IOOUT,11900) ARE                               CWC02880
        WRITE (IOOUT,11400)                                     CWC02890
IAD=1
IF (<IFOG.GE.0) IAD=3                                     CWC02900
        WRITE (IOOUT,14300) JANS(IAD),JANS(IAD+1)          CWC02910
IAD=1
IF (<IFOG.EQ.1) IAD=3                                     CWC02920
        WRITE (IOOUT,14400) JANS(IAD),JANS(IAD+1)          CWC02930
IAD=1
IF (<IFOG.EQ.2) IAD=3                                     CWC02940
        WRITE (IOOUT,14500) JANS(IAD),JANS(IAD+1)          CWC02950
IAD=1
IF (<IFOG.EQ.3) IAD=3                                     CWC02960
        WRITE (IOOUT,14600) JANS(IAD),JANS(IAD+1)          CWC02970
IAD=1
IF (<IFOG.EQ.4) IAD=3                                     CWC02980
        WRITE (IOOUT,14700) JANS(IAD),JANS(IAD+1)          CWC02990
IAD=1
IF (<IFOG.EQ.5) IAD=3                                     CWC03000
        WRITE (IOOUT,14800) JANS(IAD),JANS(IAD+1)          CWC03010
IAD=1
IF (<IPASCT(IP0)) IAD=3                                 CWC03020
        WRITE (IOOUT,12000) JP                               CWC03030
        WRITE (IOOUT,11300) VS                               CWC03040
        WRITE (IOOUT,12100) R0                               CWC03050
        IF (<IRNOT.EQ.1) WRITE (IOOUT,11500) T0           CWC03060
        IF (<IRNOT.EQ.1) WRITE (IOOUT,11600) T1           CWC03070
        IF (<IPNOT.EQ.0) GOTO 120                         CWC03080
        WRITE (IOOUT,15100)
        IF (<SLAT.GE.0.) WRITE (IOOUT,10601) SLAT          CWC03090
        SLATI=-SLAT
        IF (<SLAT.LT.0.) WRITE (IOOUT,10602) SLATI         CWC03100
        IF (<SLONG.GE.0.) WRITE (IOOUT,10701) SLONG          CWC03110
        SLONGI=-SLONG
        IF (<SLONG.LT.0.) WRITE (IOOUT,10702) SLONGI         CWC03120
        WRITE (IOOUT,10900) SJDATE                         CWC03130
        WRITE (IOOUT,11000) SZHOUR                         CWC03140
        WRITE (IOOUT,11100) C0                               CWC03150
        WRITE (IOOUT,11200) C1                               CWC03160
120  CONTINUE
        WRITE (IOOUT,15200)                                     CWC03170
        DO 150 I=1,4
        IWL=4*(I-1)
        WRITE (IOOUT,15300) (LNGTH(IWL+J),J=1,4),TR(I),T(I) CWC03180
150  CONTINUE
        WRITE (IOOUT,10000)                                     CWC03190
C* PRINT MUNITION EXPENDITURES
        WRITE (IOOUT,10130)                                     CWC03200
        WRITE (IOOUT,16200) (LNGTH(J),J=1,8)                 CWC03210
        WRITE (IOOUT,10100)                                     CWC03220
        WRITE (IOOUT,16300)                                     CWC03230
        WRITE (IOOUT,16400) X0,TIME,X0,TIME                  CWC03240
        WRITE (IOOUT,10200)                                     CWC03250
        WRITE (IOOUT,16500) ISMOKE(1),ISMOKE(1)              CWC03260
        WRITE (IOOUT,10100)                                     CWC03270
        WRITE (IOOUT,16600) IGUNK(1),IGUNK(2),IGUNK(1),IGUNK(2) CWC03280
        WRITE (IOOUT,10100)                                     CWC03290
        WRITE (IOOUT,16700)                                     CWC03300
        WRITE (IOOUT,16800)                                     CWC03310
        WRITE (IOOUT,16900) Q1(1,1,1),Y1(1,1,1),Q1(2,1,1),Y(2,1,1) CWC03320
        WRITE (IOOUT,17000) Q1(1,2,1),R1(1,1),Y(1,2,1),P(1,1,1),Q1(2,2,1), CWC03330
                                            CWC03340
                                            CWC03350
                                            CWC03360
                                            CWC03370
                                            CWC03380
                                            CWC03390
                                            CWC03400
                                            CWC03410
                                            CWC03420
                                            CWC03430
                                            CWC03440
                                            CWC03450
                                            CWC03460
                                            CWC03470
                                            CWC03480
                                            CWC03490
                                            CWC03500

```

```

*R(2,1),Y(2,2,1),P(2,1,1)          CWC03510
  WRITE(I00UT,10200)
  WRITE(I00UT,16600) IGUN(3),IGUN(4),IGUN(3),IGUN(4)    CWC03520
  WRITE(I00UT,10100)
  WRITE(I00UT,16700)
  WRITE(I00UT,16800)
  WRITE(I00UT,16900) Q1(1,1,2),Y(1,1,2),Q1(2,1,2),Y(2,1,2)    CWC03530
  WRITE(I00UT,17000) Q1(1,2,2),R(1,1),Y(1,2,2),P(1,2,1),Q1(2,2,2),    CWC03540
*R(2,1),Y(2,2,2),P(2,2,1)          CWC03550
  WRITE(I00UT,10200)
  WRITE(I00UT,16500) ISMOKE(2),ISMOKE(2)                  CWC03560
  WRITE(I00UT,10100)
  WRITE(I00UT,16600) IGUN(1),IGUN(2),IGUN(1),IGUN(2)      CWC03570
  WRITE(I00UT,10100)
  WRITE(I00UT,16700)
  WRITE(I00UT,16800)
  WRITE(I00UT,16900) Q2(1,1,1),Z(1,1,1),Q2(2,1,1),Z(2,1,1)    CWC03600
  IF(IP0 .GT. 4) R(1,2)=2.          CWC03610
  IF(IP0 .GT. 4) R(2,2)=2.          CWC03620
  WRITE(I00UT,17000) Q2(1,2,1),R(1,2),Z(1,1,1),P(1,1,2),Q2(2,2,1),    CWC03630
*R(2,2),Z(2,1,1),P(2,1,2)          CWC03640
  WRITE(I00UT,10200)
  WRITE(I00UT,16600) IGUN(3),IGUN(4),IGUN(3),IGUN(4)      CWC03650
  WRITE(I00UT,10100)
  WRITE(I00UT,16700)
  WRITE(I00UT,16800)
  WRITE(I00UT,16900) Q2(1,1,2),Z(1,2,2),Q2(2,1,2),Z(2,2,2)    CWC03660
  IF(IP0 .GT. 4) R(1,2)=1.          CWC03670
  IF(IP0 .GT. 4) R(2,2)=1.          CWC03680
  WRITE(I00UT,17000) Q2(1,2,2),R(1,2),Z(1,2,2),P(1,2,2),Q2(1,2,2),    CWC03690
*R(2,2),Z(2,2,2),P(2,2,2)          CWC03700
  WRITE(I00UT,10000)
  WRITE(I00UT,10130)
DO 7100 I=3,4                      CWC03710
IWL=4*(I-1)
J1=IWL+1
J2=IWL+4
  WRITE(I00UT,12200) (LNGTH(J), J=J1,J2)                  CWC03720
  WRITE(I00UT,10100)
  WRITE(I00UT,12300)
  WRITE(I00UT,12400) X0,TIME
  WRITE(I00UT,10200)
  WRITE(I00UT,12500) ISMOKE(2)
  WRITE(I00UT,10100)
  WRITE(I00UT,13100)
  WRITE(I00UT,13200)
  WRITE(I00UT,13300) IGUN(1),IGUN(2),Q2(I,2,1),R(I,2),P(I,1,2)    CWC03730
  WRITE(I00UT,13300) IGUN(3),IGUN(4),Q2(I,2,2),R(I,2),P(I,2,2)    CWC03740
IF (I.EQ.3) WRITE(I00UT,10140)
IF (I.EQ.4) WRITE(I00UT,10000)
7100 CONTINUE
RETURN
C* FORMAT STATEMENTS.
10000 FORMAT(1H1)                    CWC03800
10100 FORMAT(1H)                     CWC03810
10200 FORMAT(1H0)                   CWC03820
10300 FORMAT(//)                   CWC03830
10400 FORMAT(5X,21HMUNITION EXPENDITURES)    CWC03840
10500 FORMAT(5X,19HFOR HC AND WP SMOKE)     CWC03850
10601 FORMAT(45X,35HLATITUDE        - DEG   =,F7.2,6H NORTH)    CWC03860
10602 FORMAT(45X,35HLATITUDE        - DEG   =,F7.2,6H SOUTH)    CWC03870
10701 FORMAT(45X,35HLONGITUDE       - DEG   =,F7.2,6H EAST)    CWC03880
10702 FORMAT(45X,35HLONGITUDE       - DEG   =,F7.2,6H WEST)    CWC03890
10800 FORMAT(1H0,47X,36(1H*)/48X,1H*,34X,1H*/48X,1H*,4X,    CWC03900
  *26HCWIC MUNITION EXPENDITURES,4X,1H*/48X,1H*,34X,1H*/48X,1H*)    CWC04110
  *36(1H*))                  CWC04120
10900 FORMAT(45X,35HJULIAN DATE      - DAY    =,F7.0)           CWC04130
11000 FORMAT(45X,35HGMT TIME        - HOUR   =,F7.2)           CWC04140
11100 FORMAT(45X,35HCEILING         - METERS =,F7.1)           CWC04150

```

11200	FORMAT(45X,35HCLOUD COVER	- PERCENT	=,F7.1)	CWC04210
11300	FORMAT(45X,35HVISIBILITY	- KM	=,F7.3)	CWC04220
11400	FORMAT(1H0,44X,35HATMOSPHERIC EXTINCTION	CORRECTIONS	>	CWC04230
11500	FORMAT(45X,35HTEMPERATURE	- DEG C	=,F7.1)	CWC04240
11600	FORMAT(45X,35HDEW POINT	- DEG C	=,F7.1)	CWC04250
11700	FORMAT(45X,35HWIND DIRECTION	- DEG	=,F7.2)	CWC04260
11800	FORMAT(45X,35HWINDSPEED	- M/SEC	=,F7.2)	CWC04270
11900	FORMAT(45X,35HAVG ROUGHNESS ELEMENT	- CM	=,F7.1)	CWC04280
12000	FORMAT(45X,35HPASQUILL CATEGORY	-	=,5X,A2)	CWC04290
12100	FORMAT(45X,35HRELATIVE HUMIDITY	- PERCENT	=,F7.1)	CWC04300
12200	FORMAT(60X,11<1H->/60X,1H-,1X,4A2,1H-/60X,11<1H->)			CWC04310
12300	FORMAT(62X,6HLENGTH,8X,8HDURATION/62X,6HMETERS,9X,7HMINUTES)			CWC04320
12400	FORMAT(47X,6HSCREEN,8X,F7.0,9X,F7.2)			CWC04330
12500	FORMAT(59X,A2,13H SMOKE SCREEN/59X,15<1H->)			CWC04340
12600	FORMAT(59X,A2,A1,11HMM HOWITZER)			CWC04350
12700	FORMAT(47X,37HVOLLEY GUNS RATE SPACING ROUNDS)			CWC04360
12800	FORMAT(47X,30H /MIN METERS)			CWC04370
12900	FORMAT(47X,11HINITIAL: ,F5.0,6X,F8.0)			CWC04380
13000	FORMAT(47X,11HSUSTAINING: ,F5.0,F5.1,F9.0,F7.0)			CWC04390
13100	FORMAT(47X,31H ROUNDS/ RATE/ TOTAL)			CWC04400
13200	FORMAT(47X,32H 60 METERS MINUTE ROUNDS)			CWC04410
13300	FORMAT(47X,A2,A1,5HMM: ,F5.0,5X,F4.0,3X,F7.0)			CWC04420
14000	FORMAT(45X,35HSLANT RANGÉ OBS-TGT - KM	=,F7.3)		CWC04430
14100	FORMAT(45X,35HELEVATION OF TARGET - DEG	=,F7.2)		CWC04440
14200	FORMAT(45X,35HAZIMUTH OF TARGET - DEG	=,F7.2)		CWC04450
14300	FORMAT(45X,35HCORRECTED FOR VISIBILITY -	,4X,2A2)		CWC04460
14400	FORMAT(45X,35HMARITIME ARCTIC AIR MASS -	,4X,2A2)		CWC04470
14500	FORMAT(45X,35HMARITIME POLAR AIR MASS -	,4X,2A2)		CWC04480
14600	FORMAT(45X,35HCONTINENTAL POLAR AIR MASS -	,4X,2A2)		CWC04490
14700	FORMAT(45X,35HCORRECTED FOR RAIN -	,4X,2A2)		CWC04500
14800	FORMAT(45X,35HCORRECTED FOR SNOW -	,4X,2A2)		CWC04510
15000	FORMAT(1H0,44X,35HMETEOROLOGICAL INPUTS)			CWC04520
15100	FORMAT(1H0,44X,35HINPUTS FOR PASQUILL CATEGORY)			CWC04530
15200	FORMAT(1H0,44X,42HTRANSMISSION THRESHOLDS TOTAL SMOKE)			CWC04540
15300	FORMAT(45X,A42,18X,F5.3,6X,F5.3)			CWC04550
16200	FORMAT(33X,12<1H->,44X,12<1H->/33X,1H-,1X,4A2,1X,1H-,44X,1H-,1X, *4A2,1X,1H-/33X,12<1H->,44X,12<1H->)			CWC04560
16300	FORMAT(1X,2<34X,6HLENGTH,8X,8HDURATION>/1X,2<34X,6HMETERS,9X, *7HMINUTES>)			CWC04570
16400	FORMAT(1X,2<19X,6HSCREEN,8X,F7.0,9X,F7.2))			CWC04590
16500	FORMAT(32X,A2,13H SMOKE SCREEN,41X,A2,13H SMOKE SCREEN/32X, *15<1H->,41X,15<1H->)			CWC04610
16600	FORMAT(32X,A2,A1,11HMM HOWITZER,42X,A2,A1,11HMM HOWITZER)			CWC04630
16700	FORMAT(1X,2<19X,6HVOLLEY,6X,25HGUNS RATE SPACING ROUNDS))			CWC04640
16800	FORMAT(37X,4H/MIN,3X,6HMETERS,43X,4H/MIN,3X,6HMETERS)			CWC04650
16900	FORMAT(20X,11HINITIAL: ,F5.0,6X,F8.0,26X,11HINITIAL: *F5.0,6X,F8.0)			CWC04660
17000	FORMAT(1X,2<19X,11HSUSTAINING: ,F5.0,F5.1,F9.0,F7.0))			CWC04670
	END			CWC04680
				CWC04690

```

SUBROUTINE CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE,SZ HOUR,SO)          CWX00010
DIMENSION ITAB(7,9)                                                 CWX00020
COMMON /CONST/ PI,PI2,PIRAD,TWOPi,TORRMB,CDECK                  CWX00030
DATA ITAB /                                                       CWX00040
   1,1,2,3,4,6,6,                                                 CWX00050
   2,1,2,2,3,4,6,6,                                                 CWX00060
   3,1,2,3,4,4,5,6,                                                 CWX00070
   4,2,2,3,4,4,5,6,                                                 CWX00080
   5,2,2,3,4,4,4,5,                                                 CWX00090
   6,2,3,3,4,4,4,5,                                                 CWX00100
   7,3,3,4,4,4,4,5,                                                 CWX00110
   8,3,3,4,4,4,4,4,                                                 CWX00120
   9,3,4,4,4,4,4,4,                                                 CWX00130
ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))
C*METEOROLOGICAL CALCULATIONS.
C
IF(C1 .NE. 100.) GO TO 1000
IF(C0 .GT. 2133.6042) GO TO 1000
I1=0
I2=0
GO TO 2300
1000 CONTINUE
C CALCULATE ANGULAR FRACTION OF A YEAR FOR A GIVEN JULIAN DATE.
R9=PIRAD
D9=1./R9
SLAT1=SLAT*R9
A0=(SJDATE-1.)*360./365.242
C CALCULATE SOLAR DECLINATION ANGLE (A4).
A1=A0*R9
A2=279.9348+A0
A2=A2+(1.914827*SIN(A1))-(0.079525*COS(A1))
A2=A2+(0.019938*SIN(2*A1))-(0.00162*COS(2*A1))
A2=A2*R9
A3=23.4438*R9
A4=SIN(A3)*SIN(A2)
A4=ASIN(A4)
C CALCULATE THE TIME OF MERIDIAN PASSAGE - TRUE SOLAR NOON (A5).
A5=12.+(<0.12357*SIN(A1))-(<0.004289*COS(A1))
A5=A5+(<0.153809*SIN(2*A1))+(<0.060783*COS(2*A1))
C CALCULATE SOLAR HOUR ANGLE (A6)*** NOTE THIS VERSION USES + SIGN
C ON SLONG DUE TO EAST-LONGITUDE POSITIVE CONVENTION.
A6=15.*(<SZ HOUR-A5)+SLONG
A6=A6*R9
C CALCULATE SOLAR ALTITUDE (A7).
A7=SIN(SLAT1)*SIN(A4)+COS(SLAT1)*COS(A4)*COS(A6)
A7=ASIN(A7)
1100 CONTINUE
A7=A7*D9
C CALCULATE INSOLATION CLASS NUMBER.
I2=0
IF(A7 .LE. 60.) GO TO 1200
I2=4
GO TO 1500
1200 CONTINUE
IF(A7 .LE. 35.) GO TO 1300
I2=3
GO TO 1500
1300 CONTINUE
IF(A7 .LE. 15.) GO TO 1400
I2=2
GO TO 1500
1400 CONTINUE
IF(A7 .LE. 0.) GO TO 2200
I2=1
C CALCULATE NET RADIATION INDEX FOR DAYTIME.
1500 CONTINUE
I3=0
IF(C1 .GT. 50.) GO TO 1600
I3=I2
GO TO 1900

```

```

1600 CONTINUE
IF(C0 .GT. 2133.6042) GO TO 1700
I3=I2-2
GO TO 1900
1700 CONTINUE
IF(C0 .GE. 4876.8096) GO TO 1800
I3=I2-1
GO TO 1900
1800 CONTINUE
IF(C1 .NE. 100.) GO TO 1900
I3=I2-1
1900 CONTINUE
IF(I3 .NE. 0) GO TO 2000
I3=I2
2000 CONTINUE
IF(I3 .GT. 1) GO TO 2100
I3=i
2100 CONTINUE
I1=I3
GO TO 2300
C COMPUTE NET RADIATION INDEX FOR NIGHTTIME.
2200 CONTINUE
IF(C1 .GT. 40.) GO TO 2250
I1=-2
GO TO 2300
2250 CONTINUE
I1=-1
C CALCULATE PASQUILL STABILITY CATALOGY.
2300 CONTINUE
I4=0
I5=0
IF(I1 .NE. 4) GO TO 2400
I4=1
2400 CONTINUE
IF(I1 .NE. 3) GO TO 2420
I4=2
2420 CONTINUE
IF(I1 .NE. 2) GO TO 2440
I4=3
2440 CONTINUE
IF(I1 .NE. 1) GO TO 2460
I4=4
2460 CONTINUE
IF(I1 .NE. 0) GO TO 2480
I4=5
2480 CONTINUE
IF(I1 .NE. -1) GO TO 2500
I4=6
2500 CONTINUE
IF(I1 .NE. -2) GO TO 2520
I4=7
2520 CONTINUE
IF(S0 .GE. 2.) GO TO 2540
I5=1
GO TO 2700
2540 CONTINUE
IF(S0 .GE. 4.) GO TO 2560
I5=2
GO TO 2700
2560 CONTINUE
IF(S0 .GE. 6.) GO TO 2580
I5=3
GO TO 2700
2580 CONTINUE
IF(S0 .GE. 7.) GO TO 2600
I5=4
GO TO 2700
2600 CONTINUE
IF(S0 .GE. 8.) GO TO 2620
I5=5

```

CWX00700
CWX00710
CWX00720
CWX00730
CWX00740
CWX00750
CWX00760
CWX00770
CWX00780
CWX00790
CWX00800
CWX00810
CWX00820
CWX00830
CWX00840
CWX00850
CWX00860
CWX00870
CWX00880
CWX00890
CWX00900
CWX00910
CWX00920
CWX00930
CWX00940
CWX00950
CWX00960
CWX00970
CWX00980
CWX00990
CWX01000
CWX01010
CWX01020
CWX01030
CWX01040
CWX01050
CWX01060
CWX01070
CWX01080
CWX01090
CWX01100
CWX01110
CWX01120
CWX01130
CWX01140
CWX01150
CWX01160
CWX01170
CWX01180
CWX01190
CWX01200
CWX01210
CWX01220
CWX01230
CWX01240
CWX01250
CWX01260
CWX01270
CWX01280
CWX01290
CWX01300
CWX01310
CWX01320
CWX01330
CWX01340
CWX01350
CWX01360
CWX01370
CWX01380
CWX01390

	GO TO 2700	CWX01400
2620	CONTINUE	CWX01410
	IF(S0 .GE. 10.) GO TO 2640	CWX01420
	I5=6	CWX01430
	GO TO 2700	CWX01440
2640	CONTINUE	CWX01450
	IF(S0 .GE. 11.) GO TO 2660	CWX01460
	I5=7	CWX01470
	GO TO 2700	CWX01480
2660	CONTINUE	CWX01490
	IF(S0 .GE. 12.) GO TO 2680	CWX01500
	I5=8	CWX01510
	GO TO 2700	CWX01520
2680	CONTINUE	CWX01530
	I5=9	CWX01540
2700	CONTINUE	CWX01550
	IPO=ITAB(I4,I5)	CWX01560
	RETURN	CWX01570
	END	CWX01580

```

SUBROUTINE CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U V,Y1)          CWY00010
DIMENSION A(6),S(6,3),D(6,3),H(2,2),U(2,2),V(2)                  CWY00020
DIMENSION W(6)                                                       CWY00030
COMMON /CONST/ PI,PI2,PIRAD,TWOPi,TORRMB,CDEGK                 CWY00040
DATA A/0.4,0.32,0.22,0.144,0.102,0.076/                           CWY00050
DATA S/                                                        CWY00060
*.139085297,.122097643,.110104377,.097649832,.070772166,.055487093,CWY00070
*.015017284,.010970370,.010962963,.010418519,7.27284E-3,6.55309E-3,CWY00080
*-1.02581E-4,-6.80135E-5,-6.73401E-5,-6.83502E-5,-4.50056E-5,CWY00090
*-4.01796E-5/                                                 CWY00100
DATA D/                                                        CWY00110
*.944814815,.894803591,.854792368,.816026936,.786026936,.726015713,CWY00120
*-4.85185E-3,-4.83951E-3,-4.82716E-3,-6.07407E-3,-6.07407E-3,CWY00130
*-6.06173E-3/                                                 CWY00140
*3.7037E-5,3.59147E-5,3.47924E-5,4.7138E-5,4.7138E-5,4.60157E-5/CWY00150
DATA W/0.016,0.016,0.016,0.016,0.016,0.016/                      CWY00160
C*ATMOSPHERIC DIFFUSION CALCULATIONS.                            CWY00170
A1=-1.24+1.19* ALOG10(ARE)                                       CWY00180
Z1=10.**A1                                                       CWY00190
A2=ABS(DLS-D0)*(PI/180.)                                         CWY00200
R2=SQR((13.69/(13.69*SIN(A2)*SIN(A2)+COS(A2)*COS(A2))))      CWY00210
Y1=1.09521547+(0.02906894*R0)-(4.9575E-04*R0*R0)+           CWY00220
(4.82E-06*R0*R0*R0)                                              CWY00230
2 Y2=3.364059144+(0.060502571*R0)-(1.15301E-03*R0*R0)+        CWY00240
(1.33942E-05*R0*R0*R0)                                           CWY00250
2 C2=S(IP0,1)+S(IP0,2)*Z1+S(IP0,3)*Z1**2                     CWY00260
D1=D(IP0,1)+D(IP0,2)*Z1+D(IP0,3)*Z1**2                     CWY00270
D2=1/D1                                                       CWY00280
DO 5400 I=1,4                                                 CWY00290
C*CALCULATE CROSSWIND INTEGRATED CONCENTRATION FOR WP SMOKE.
DO 5300 K=1,2                                                 CWY00300
IF (I.LT. 3 .AND. IP0 .GT. 4) GOTO 5300                         CWY00310
S1=U(K,1)+0.74*A(IP0)*100.**0.9                                CWY00320
S2=U(K,2)+0.667*C2*100.**D1                                     CWY00330
V(K)=(U(IP0)*Y2*H(K,2))/(PI*S1*S2)                               CWY00340
5300 CONTINUE                                                 CWY00350
5400 CONTINUE                                                 CWY00360
      RETURN                                                 CWY00370
      END                                                 CWY00380
                                         CWY00390

```

```

SUBROUTINE CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0) CWZ00010
  REAL ME(2)
  DIMENSION C(4,2),H(2,2),P(4,2,2),Q1(4,2,2),Q2(4,2,2),R(4,2),V(2),
  *Y(4,2,2),Z(4,2,2)
  DATA ME/0.4,0.4/
  SUBROUTINE CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0) CWZ00020
  C*MUNITIONS EXPENDITURES.
  DO 6900 I=1,4 CWZ00030
  C* CALCULATE INITIAL SHELL SPACING FOR HC SMOKE CWZ00040
  DO 6800 K=1,2 CWZ00050
  IF(I .GT. 2) GO TO 6100 CWZ00060
  Y(I,1,K)=45.*S3 CWZ00070
  C* CALCULATE SUSTAINING SHELL SPACING FOR HC SMOKE CWZ00080
  IF(C(I,1) .NE. 0.) GO TO 5500 CWZ00090
  Y(I,2,K)=0. CWZ00100
  GO TO 5600 CWZ00110
  5500 CONTINUE CWZ00120
  Y(I,2,K)=1/R2*((0.731*ME(K)*Y1*H(K,1))/(C2*S3+C(I,1))**D2) CWZ00130
  IF(Y(I,2,K) .GT. X0) Y(I,2,K)=X0 CWZ00140
  5600 CONTINUE CWZ00150
  IF(Y(I,2,K) .NE. 0.) GO TO 5700 CWZ00160
  Q1(I,1,K)=1. CWZ00170
  Q1(I,2,K)=1. CWZ00180
  GO TO 5900 CWZ00190
  5700 CONTINUE CWZ00200
  C* CALCULATE INITIAL VOLLEY FOR HC SMOKE CWZ00210
  IF(Y(I,1,K) .GT. Y(I,2,K)) Y(I,1,K)=Y(I,2,K) CWZ00220
  Q1(I,1,K)=X0/Y(I,1,K) CWZ00230
  Q5=AIINT(Q1(I,1,K)) CWZ00240
  Q6=Q1(I,1,K)-Q5 CWZ00250
  Q6=Q1(I,1,K)-Q5 CWZ00260
  IF(Q6 .EQ. 0.) GO TO 5900 CWZ00270
  Q1(I,1,K)=Q5+1. CWZ00280
  5800 CONTINUE CWZ00290
  C* CALCULATE NUMBER OF GUNS FOR SUSTAINING VOLLEYS (HC) CWZ00300
  Q1(I,2,K)=X0/Y(I,2,K) CWZ00310
  Q5=AIINT(Q1(I,2,K)) CWZ00320
  Q6=Q1(I,2,K)-Q5 CWZ00330
  IF(Q6 .EQ. 0.) GO TO 5900 CWZ00340
  Q1(I,2,K)=Q5+1. CWZ00350
  5900 CONTINUE CWZ00360
  C* CALCULATE RATE OF FIRE FOR HC SMOKE CWZ00370
  R(I,1)=0.5 CWZ00380
  IF(C(I,1) .NE. 0.) GO TO 6000 CWZ00390
  R(I,1)=0. CWZ00400
  6000 CONTINUE CWZ00410
  C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (HC SMOKE) CWZ00420
  P(I,K,1)=Q1(I,1,K)+0.5*Q1(I,2,K)*(TIME-2.) CWZ00430
  Q5=AIINT(P(I,K,1)) CWZ00440
  Q6=P(I,K,1)-Q5 CWZ00450
  IF(Q6 .EQ. 0.) GO TO 6100 CWZ00460
  P(I,K,1)=Q5+1. CWZ00470
  6100 CONTINUE CWZ00480
  IF(I .LT. 3 .AND. IP0 .GT. 4) GO TO 6775 CWZ00490
  C* SHELL SPACING (Z( )) AND VOLLEYS (Q( )) - WP SMOKE CWZ00500
  IF(C(I,2) .NE. 0.) GO TO 6200 CWZ00510
  Z(I,1,K)=0. CWZ00520
  Z(I,2,K)=0. CWZ00530
  Q2(I,1,K)=0. CWZ00540
  Q2(I,2,K)=0. CWZ00550
  GO TO 6400 CWZ00560
  6200 CONTINUE CWZ00570
  IF(I .LT. 3) GO TO 6250 CWZ00580
  IF(I .GT. 2) Q2(I,1,K)=0.6*C(I,2)/V(K) CWZ00590
  GO TO 6300 CWZ00600
  6250 CONTINUE CWZ00610
  Z(I,1,K)=Y(K)/C(I,2)*100. CWZ00620
  Z(I,2,K)=Z(I,1,K) CWZ00630
  Q2(I,1,K)=X0/Z(I,2,K)+1. CWZ00640
  6300 CONTINUE CWZ00650
  Q5=AIINT(Q2(I,1,K)) CWZ00660
  Q6=Q2(I,1,K)-Q5 CWZ00670
  C* CALCULATE NUMBER OF GUNS FOR SUSTAINING VOLLEYS (WP) CWZ00680
  Q1(I,2,K)=X0/Y(I,2,K) CWZ00690
  Q5=AIINT(Q1(I,2,K)) CWZ00700

```

```

IF (Q6 .EQ. 0.) GO TO 6350
Q2(I,1,K)=Q5+1.
6350 CONTINUE
Q2(I,2,K)=Q2(I,1,K)
6400 CONTINUE
C* RATE OF FIRE FOR WP SMOKE
IF(C(I,2) .NE. 0.) GO TO 6425
R(I,2)=0.
GO TO 6600
6425 CONTINUE
IF (I .GT. 2) GOTO 6450
R(I,2)=(Z(I,2,K)+60.)/S3
GO TO 6475
6450 CONTINUE
R(I,2)=120./S3
6475 CONTINUE
R(I,2)=R(I,2)/20.
R5=AINTR(R(I,2))
R6=R(I,2)-R5
IF(R6 .LT. 0.5) GO TO 6500
R5=R5+1.
6500 CONTINUE
IF(R5 .NE. 0.) GO TO 6550
R5=1.
6550 CONTINUE
R(I,2)=R5*20./60.
6600 CONTINUE
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (WP)
IF(C(I,2) .NE. 0.) GO TO 6650
P(I,K,2)=0.
GO TO 6700
6650 CONTINUE
IF(I .GT. 2) GO TO 6700
P(I,K,2)=Q2(I,2,K)*(1./R(I,2))*(TIME-R(I,2))
GO TO 6750
6700 CONTINUE
P(I,K,2)=Q2(I,1,K)*(X0/60.+1.)*(1./R(I,2))*(TIME-R(I,2))
6750 CONTINUE
Q5=AINTR(P(I,K,2))
Q6=P(I,K,2)-Q5
IF(Q6 .EQ. 0.) GO TO 6800
P(I,K,2)=Q5+1.
GO TO 6800
6775 CONTINUE
C* CALCULATIONS FOR E AND F STABILITY CAT (STABLE FLOW)
C* INITIAL SHELL SPACING FOR WP SMOKE
IF(I .EQ. 1 .AND. K .EQ. 1) Z(I,1,K)=100.
IF(I .EQ. 2 .AND. K .EQ. 1) Z(I,1,K)=50.
IF(I .LT. 3 .AND. K .EQ. 2) Z(I,1,K)=100.
C* SUSTAINING SHELL SPACING FOR WP SMOKE
IF(I .EQ. 1 .AND. K .EQ. 1) Z(I,2,K)=100.
IF(I .EQ. 1 .AND. K .EQ. 2) Z(I,2,K)=200.
IF(I .EQ. 2 .AND. K .EQ. 1) Z(I,2,K)=50.
IF(I .EQ. 2 .AND. K .EQ. 2) Z(I,2,K)=100.
C* INITIAL VOLLEY FOR WP SMOKE
Q2(I,1,K)=X0/Z(I,1,K)+1.
C* SUSTAINING VOLLEY FOR WP SMOKE
Q2(I,2,K)=X0/Z(I,2,K)+1.
C* RATE OF FIRE FOR WP SMOKE
IF(K .EQ. 1) R(I,2)=.5
IF(K .EQ. 2) R(I,2)=1.
C* TOTAL NUMBER OF ROUNDS REQUIRED (WP)
P(I,K,2)=Q2(I,1,K)+Q2(I,2,K)*1./R(I,2)*(TIME-R(I,2))
6800 CONTINUE
R(I,2)=1./R(I,2)
6900 CONTINUE
RETURN
END

```

CWZ00710
CWZ00720
CWZ00730
CWZ00740
CWZ00750
CWZ00760
CWZ00770
CWZ00780
CWZ00790
CWZ00800
CWZ00810
CWZ00820
CWZ00830
CWZ00840
CWZ00850
CWZ00860
CWZ00870
CWZ00880
CWZ00890
CWZ00900
CWZ00910
CWZ00920
CWZ00930
CWZ00940
CWZ00950
CWZ00960
CWZ00970
CWZ00980
CWZ00990
CWZ01000
CWZ01010
CWZ01020
CWZ01030
CWZ01040
CWZ01050
CWZ01060
CWZ01070
CWZ01080
CWZ01090
CWZ01100
CWZ01110
CWZ01120
CWZ01130
CWZ01140
CWZ01150
CWZ01160
CWZ01170
CWZ01180
CWZ01190
CWZ01200
CWZ01210
CWZ01220
CWZ01230
CWZ01240
CWZ01250
CWZ01260
CWZ01270
CWZ01280
CWZ01290
CWZ01300
CWZ01310
CWZ01320
CWZ01330
CWZ01340
CWZ01350
CWZ01360
CWZ01370
CWZ01380

SUBROUTINE ITAM(IERR,TFNL)

THIS ROUTINE IS AN INVERSION OF THE NV&EOL TARGET ACQUISITION MODEL, WITH EMPHASIS ON THE DEGRADATION (TRANSMITTANCE) REQUIRED TO PREVENT DETECTION ABOVE A GIVEN PROBABILITY.

INPUTS ARE ON STANDARDIZED RECORDS:

KEY WORD - COLUMNS 1-4, FROM AMONG TARV, SENS, GO, PAGE AND DONE. DATA FIELDS, ALL REAL - COLS 11-20, 21-30, 31-40, ..., 71-80.

INPUT CARDS ARE ORDER INDEPENDENT, WITH ARBITRARY OR SYSTEM DEFAULTS CHOSEN IF CARDS ARE LEFT OUT, OR IF VALUES ARE INPUT 0. THE EXCEPTION TO ORDER INDEPENDENCE IS THAT AFTER EACH SET OF INPUTS A <GO> CARD MUST BE PLACED TO INITIATE EXECUTION OF ONE LOOP THROUGH THE PROGRAM. FOLLOWING, OR IN PLACE OF THE LAST INPUT-SET GO CARD, A <DONE> CARD MUST BE PROVIDED TO CAUSE CONTROL TO EXIT ITAM AND RETURN TO THE SCREEN EXEC MODULE.

IN SUBSEQUENT RUNS (DELINEATED BY GO CARDS) TABLES MAY BE PRODUCED LINE-BY-LINE, AND ANY INPUT WHICH HAS NOT BEEN CHANGED (IE INPUT AS 0.) WILL USE THE VALUE GIVEN ON THE PREVIOUS RUN. EXCEPTIONS ARE THE FOV AND AMAG VALUES WHICH MUST BE SPECIFIED WHENEVER A NEW DEVICE NUMBER (LSC) IS INPUT, OR THESE VALUES ASSUME DEFAULTS FOR THE DEVICE.

KEY WORD	COLS.	VARIABLE	DESCRIPTION	
TARV	1-4 11-20	ACON	TARGET/SCENARIO DESCRIPTION - INTRINSIC CONTRAST (DIMENSIONLESS) OR TEMPERATURE DIFFERENCE OF TARGET/BACKGROUND (DEG K) FOR THERMAL DEVICES.	ITAM0010 ITAM0020 ITAM0030 ITAM0040 ITAM0050 ITAM0060 ITAM0070 ITAM0080 ITAM0090 ITAM0100 ITAM0110 ITAM0120 ITAM0130 ITAM0140 ITAM0150 ITAM0160 ITAM0170 ITAM0180 ITAM0190 ITAM0200 ITAM0210 ITAM0220 ITAM0230 ITAM0240 ITAM0250 ITAM0260 ITAM0270 ITAM0280 ITAM0290 ITAM0300 ITAM0310 ITAM0320 ITAM0330 ITAM0340 ITAM0350 ITAM0360 ITAM0370 ITAM0380 ITAM0390 ITAM0400 ITAM0410 ITAM0420 ITAM0430 ITAM0440 ITAM0450 ITAM0460 ITAM0470 ITAM0480 ITAM0490 ITAM0500 ITAM0510 ITAM0520 ITAM0530 ITAM0540 ITAM0550 ITAM0560 ITAM0570 ITAM0580 ITAM0590 ITAM0600 ITAM0610 ITAM0620 ITAM0630 ITAM0640 ITAM0650 ITAM0660 ITAM0670 ITAM0680 ITAM0690 ITAM0700
SENS	1-4 11-20 21-30 31-40 41-50 51-60 61-70	SOG R TARSZ ZONE ALFLG	SKY/GROUND RATIO. (DIMENSIONLESS), USED FOR NON-THERMAL DEVICES. SOG SHOULD INCLUDE A FACTOR FOR CLOUD REFLECTANCE - RANGE TO TARGET (KM) - TARGET MINIMUM DIMENSION (M). - SEARCH ZONE (DEG.**2) - AMBIENT ILLUM. CATEGORY (SEE LIST BELOW) CONVERTED TO NAL, AN INTEGER, IN PROGRAM. USED FOR NON-THERMAL DEVICES. - IF ALFLG=0., THE USER MUST PROVIDE AN AMBIENT ILLUM. HERE (FT. COLS.)	ITAM0350 ITAM0360 ITAM0370 ITAM0380 ITAM0390 ITAM0400 ITAM0410 ITAM0420 ITAM0430 ITAM0440 ITAM0450 ITAM0460 ITAM0470 ITAM0480 ITAM0490 ITAM0500 ITAM0510 ITAM0520 ITAM0530 ITAM0540 ITAM0550 ITAM0560 ITAM0570 ITAM0580 ITAM0590 ITAM0600 ITAM0610 ITAM0620 ITAM0630 ITAM0640 ITAM0650 ITAM0660 ITAM0670 ITAM0680 ITAM0690 ITAM0700
PAGE	1-4		(OPTIONAL) - FORCE PAGE EJECT, WRITE NEW HEADER. (USEFUL FOR TABLE GENERATION.)	ITAM0630 ITAM0640 ITAM0650 ITAM0660 ITAM0670 ITAM0680 ITAM0690 ITAM0700
GO	1-4		EXECUTE ONE LOOP WITH GIVEN INPUTS	ITAM0670 ITAM0680 ITAM0690 ITAM0700
DONE	1-4		END COMPUTATIONS AND EXIT THE ITAM	ITAM0700

ROUTINE. (MAY BE USED IN PLACE OF ITA00710
OR FOLLOWING THE LAST GO CARD)

OUTPUTS: TFNL, THE TOTAL PATH TRANSMITTANCE (DIMENSIONLESS)
REQUIRED TO KEEP THE PROBABILITY OF DETECTION LESS THAN OR EQUAL TO THAT INPUT, AS COMPUTED FROM THE FINAL SET OF INPUTS PRECEDING THE DONE CARD.

PRINT OUTPUTS: ALL INPUT PARAMETERS AND
C = CONTRAST OR TEMPERATURE DIFFERENCE AT DEVICE
REQUIRED FOR GIVEN LEVEL OF DETECTION PROBABILITY
RC = RESOLVABLE CYCLES REQUIRED AT TARGET FOR GIVEN
PROBABILITY OF DETECTION.
TTOT = TOTAL PATH TRANSMITTANCE TO REDUCE ACON AT TARGET
TO C AT DEVICE.

SUBROUTINES: ITAM CALLED BY SCREEN. ITAM CALLS CYCLE, CINV, TREQ.

AMBIENT ILLUMINATION CATEGORIES:

ALFLG OR NAL	FT. CDLS.	APPROX. CONDITIONS
0	USER SPEC.	
1	100.	CLEAR TO SLIGHT OVERCAST
2	10.	HEAVILY OVERCAST DAY
3	1.	SUNSET
4	0.1	1/4 HR. AFTER SUNSET
5	0.01	1/2 HR. AFTER SUNSET
6	0.001	MOONLIT, CLEAR
7	0.0001	MOONLIT, OVERCAST
8	0.00001	STARLIGHT ONLY
9	0.000001	MINIMAL STARLIGHT

DIMENSION RV(7), IT(2), AMAGW(14), AMAGNC(14), FOVW(14), FOVN(14),
* ALFA(9), IR(10), BETA(9), GAMA(9), AMRAL(6,3), AMRGM(6,3)
COMMON /IOUNIT/IOUTN, IOUTI, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUITA01070
COMMON /GEOMET/PTS(15), IGEOSW
DATA IR/2HDO, 2HNE, 2HGO, 2H , 2HTA, 2HRV, 2HSE, 2HNS, 2HPA, 2HGE/
DATA AMAGW/1.0, 7.0, 12*0./,
* AMAGN/1.0, 7.0, 12*0./,
* FOVW/24.5, 8.0, 15.0, 9.0, 6.0, 4.67, 4.67, 10.62, 9.24, 8.0, 0.4*0./,
* FOVN/24.5, 8.0, 3*0., 1.56, 0., 3.54, 2.31, 1.0, 0.4*0./,
DATA TGT, SGR, FV, RN, ALV, DMG, AJB, PD, DVN, DMD, ACN/2.3,
* 3./? .1. .1. .1. .1. .2. .1. .1. /
DATA ALFA/1.089005, .009730, 1.8528, -0.0002197, 139.0, 0.0, 0.0, 0.0/
DATA BETA/-1.801654, .03536, -3.13169, .0514, 0., .0651, .1124, .0968,
* 12/
DATA GAMA/4.217775, 2.3575, 5.17227, 2.32178, 2.7933, 1.6149, 1.3364,
*.56867, .29/
DATA AMRAL/.05850156, .1022697, .2444688, .7264854, 3.736508,
*-16.87941, .06709132, .1137755, .3177794, .7642010, 6.549327,
*-16.18440, .04109820, .0715830, .1524969, .2319469, 1.494814,
*.5.731956/
DATA AMRGM/2.94273, 2.43789, 2.11113, 2.05233, 3.33957, -4.98529,
* 5.22111, 4.28211, 4.13668, 3.44685, 9.16003, -7.70891,
* 2.80725, 2.70329, 2.48447, 1.63044, 2.73650, 3.47838 /
C ***** INITIALIZE VALUES (FIRST EXECUTION)
PB=.632
ACON=0.
SGG=0.
RNG=0.
TARS2=0.
UZONE=0.
ALFLG=0.
ALIGHT=0.
PDET=0.
DYNUM=0.
DMODE=0.
DFOV=0.

```

DMAG=0. ITA01410
AJOB=0. ITA01420
TBAR=0. ITA01430
NAL=0. ITA01440
TFNL=1. ITA01450
ILAST=0. ITA01460
IFLGFV=0. ITA01470
IFLCMC=0. ITA01480
ITA01490
C ***** OUTPUT LINE COUNTER ITA01500
ILINE=0 ITA01510
C ***** INITIALIZE VALUES (REPEATED EXECUTIONS) ITA01520
C ***** INPUT CARD COUNTER ITA01530
4 ICOV=0 ITA01540
IF (ILAST.NE.0) GOTO 6 ITA01550
C ***** FLAG TO WRITE TITLE ITA01560
ITITL=0 ITA01570
TTOT=1. ITA01580
RC=0. ITA01590
C=0. ITA01600
ICM=0 ITA01610
INEWD=0 ITA01620
ITA01630
C ***** READ A CARD ITA01640
5 READ(IUIN,900) IT(1),IT(2),(RV(I),I=1,7) ITA01650
900 FORMAT(2A2,6X,7F10.2) ITA01660
ICOV=ICOV+1 ITA01670
IF (IT(1).EQ.IR(1).AND.IT(2).EQ.IR(2)) GO TO 6 ITA01680
IF (IT(1).EQ.IR(3).AND.IT(2).EQ.IR(4)) GO TO 9 ITA01690
IF (IT(1).EQ.IR(5).AND.IT(2).EQ.IR(6)) GO TO 7 ITA01700
IF (IT(1).EQ.IR(7).AND.IT(2).EQ.IR(8)) GO TO 8 ITA01710
IF (IT(1).EQ.IR(9).AND.IT(2).EQ.IR(10)) GOTO 2 ITA01720
WRITE(CDOUT,901) IT(1),IT(2),(RV(I),I=1,7) ITA01730
901 FORMAT(1X,61HTHE FOLLOWING CARD DOES NOT CONFORM TO ITAM INPUT CONITA01740
*VENTIONS/1X,2A2,6X,7E10.3) ITA01750
IF (ICOV.LE.5) GO TO 5 ITA01760
IERR=1 ITA01770
GOTO 1 ITA01780
C ***** ALL DONE ITA01790
6 IF (ICOV.GT.1) GOTO 3 ITA01800
1 WRITE(CDOUT,902) TFNL ITA01810
902 FORMAT(1H0,5X,41H*** FINAL TOTAL TRANSMISSION FROM ITAM = ,F5.3) ITA01820
RETURN ITA01830
C ***** TARGET CARD PROCESSING (TARV) ITA01840
7 RCHK=ACON ITA01850
IF (RV(1).NE.0.) ACON=RV(1) ITA01860
IF (ACON.NE.RCHK) ITITL=1 ITA01870
IF (RV(2).NE.0.) SOG=RV(2) ITA01880
IF (RV(3).NE.0.) RNG=RV(3) ITA01890
IF (ICEOSW.NE.1) GO TO 477 ITA01900
477 RNG=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+(PTS(3)-PTS(6))**2) ITA01910
CONTINUE ITA01920
RCHK=TARSZ ITA01930
IF (RV(4).NE.0.) TARSZ=RV(4) ITA01940
IF (TARSZ.NE.RCHK) ITITL=1 ITA01950
RCHK=UZONE ITA01960
IF (RV(5).NE.0.) UZONE=RV(5) ITA01970
IF (RCHK.NE.UZONE) ITITL=1 ITA01980
IF (RV(6).NE.0.) ALFLG=RV(6) ITA01990
IF (RV(7).NE.0.) ALIGHT=RV(7)
GO TO 5 ITA02000
C ***** SENSOR CARD PROCESSING (SENS) ITA02010
8 IF (RV(1).GT.0.) PDET=RV(1) ITA02020
RCHK=DNUM ITA02030
IF (RV(2).NE.0.) DNUM=RV(2) ITA02040
IF (DNUM.NE.RCHK) ITITL=1 ITA02050
IF (DNUM.NE.RCHK) INEWD=1 ITA02060
RCHK=DMODE ITA02070
IF (RV(3).NE.0.) DMODE=RV(3) ITA02080
IF (DMODE.NE.RCHK) ITITL=1 ITA02090
IF (INEWD.EQ.1) DFOV=0. ITA02100
IF (INEWD.EQ.1) IFLGFV=0

```

```

RCHK=DFOV
IF<(RV(4),NE,0.)> DFOV=RV(4)
IF<(RV(4),NE,0.,AND,IFLGFV,EQ,1)> IFLGFV=0
IF<(DFOV,NE,RCHK)> ITITL=1
IF<(INEW0,EQ,1)> DMAG=0.
IF<(INEW0,EQ,1)> IFLCMG=0
RCHK=DMAG
IF<(RV(5),NE,0.)> DMAG=RV(5)
IF<(RV(5),NE,0.,AND,IFLCMG,EQ,1)> IFLCMG=0
IF<(DMAG,NE,RCHK)> ITITL=1
RCHK=AJOB
IF<(RV(6),NE,0.)> AJOB=RV(6)
IF<(AJOB,NE,RCHK)> ITITL=1
GO TO 5
C ***** PAGE TO NEW HEADER
2 ITITL=1
GOTO 5
C ***** BEGIN PROCESSING <GO>
C ***** FIRST CHECK IF DEFAULTS NEEDED, AND SET INTEGER VALUES, LIGHT CA
339 ILAST=1
RCHK=DVNUM
IF<(DVNUM,EQ,0.)> DVNUM=DVN
IF<(RCHK,NE,DVNUM)> ITITL=1
LSC=IFIX(DVNUM+.0001)
IF<(LSC,LT,1)> LSC=1
IF<(LSC,GT,14)> LSC=14
RCHK=DMODE
IF<(DMODE,EQ,0.)> DMODE=DMD
IF<(RCHK,NE,DMODE)> ITITL=1
MODE=IFIX(DMODE+.0001)
IF<(MODE,LT,1)> MODE=1
IF<(MODE,GT,2)> MODE=2
IF<(PDET,EQ,0.)> PDET=PD
PS=PDET
IF<(PS,GT,1.)> PS=1.
IF<(PS,LT,0.)> PS=0;
RCHK=DFOV
IF<(DFOV,LE,0.0,AND,MODE,EQ,1)> DFOV=F0VW(LSC)
IF<(DFOV,LE,0.0,AND,MODE,EQ,2)> DFOV=F0VN(LSC)
IF<(DFOV,LE,0.0)> IFLGFV=1
IF<(DFOV,LE,0.,AND,MODE,EQ,1)> DFOV=FV
IF<(DFOV,LE,0.,AND,MODE,EQ,2)> DFOV=FV/2.
IF<(RCHK,NE,DFOV)> ITITL=1
FOV=DFOV
RCHK=DMAG
IF<(DMAG,LE,0.0,AND,MODE,EQ,1)> DMAG=AMAGW(LSC)
IF<(DMAG,LE,0.0,AND,MODE,EQ,2)> DMAG=AMAGN(LSC)
IF<(DMAG,LE,0.0)> IFLGMG=1
IF<(DMAG,LE,0.0)> DMAG=DMD
IF<(RCHK,NE,DMAG)> ITITL=1
AMAG=DMAG
RCHK=AJOB
IF<(AJOB,EQ,0.0)> AJOB=AJB
IF<(RCHK,NE,AJOB)> ITITL=1
RCHK=ACON
IF<(ACON,EQ,0.0)> ACON=ACN
IF<(RCHK,NE,ACON)> ITITL=1
C=ACON
IF<(SGG,EQ,0.0)> SGG=SGR
IF<(RNG,LE,0.0)> RNG=RN
R=RNG
RCHK=TARSZ
IF<(TARSZ,LE,0.0)> TARSZ=TGT
IF<(RCHK,NE,TARSZ)> ITITL=1
DIM=TARSZ
RCHK=UZONE
IF<(UZONE,LE,0.0)> UZONE=F0V**2
IF<(RCHK,NE,UZONE)> ITITL=1
ZONE=UZONE
IF<(ALFLG,LT,0.)> ALFLG=0.

```

```

IF (<ALFLG.GT.9.) ALFLG=9. ITA02810
IAL=IFIIX(ALFLG+0,0001) ITA02820
IF(<IAL.GT.0) AL=10.**(3-IAL)+1.E-10 ITA02830
IF(<IAL.EQ.0.AND.ALIGHT.EQ.0.0) ALIGHT=ALV ITA02840
IF(<IAL.EQ.0) AL=ALIGHT ITA02850
C ***** CHECK AND WRITE PAGE TITLE HEADING ITA02860
10 ILINE=ILINE+1 ITA02870
IF(<ITITL.EQ.0.AND.ILINE.LE.48) GO TO 11 ITA02880
IF (<ITITL.EQ.0.AND.ILINE.GT.49) WRITE (100UT,905) ITA02890
905 FORMAT(1H,35X,28H** CONTINUED ON NEXT PAGE **) ITA02900
WRITE(100UT,910) ITA02910
910 FORMAT(1H0/,41X,43(1H*)/41X,1H*,41X,1H*/41X,1H*,2X,37HINVERSE SITA02920
*STATIC TARGET DETECTION MODEL,2X,1H*/41X,1H*,41X,1H*/41X,43(1H*)) ITA02930
WRITE(100UT,911) ITA02940
911 FORMAT(1H0,64X,28HTARGET INTRINSIC CONTRAST OR) ITA02950
WRITE(100UT,912) LSC,ACON ITA02960
912 FORMAT(20X,13HDEVICE NUMBER,4X,I2,32X,22HTEMPERATURE DIFFERENCE, ITA02970
* 2X,F7.3) ITA02980
IF(MODE.EQ.2) WRITE(100UT,913) DIM ITA02990
913 FORMAT(1H0,19X,11HF0V TYPE -,4X,6HNARROW,24X,28HMINIMUM TARGET DI ITA03000
*MENSION (M),2X,F7.3) ITA03010
IF(MODE.EQ.1) WRITE(100UT,914)DIM ITA03020
914 FORMAT(1H0,19X,11HF0V TYPE -,6X,4HWIDE,24X,28HMINIMUM TARGET DIME ITA03030
*MNSION (M),2X,F7.3) ITA03040
IF(<IFLGFY.EQ.0) WRITE(100UT,915) FOV,AJOB ITA03050
915 FORMAT(1H0,19X,9HF0V (DEG),5X,F7.3,24X,27HACQUISITION LEVEL (50 PC ITA03060
*NT),3X,F7.3) ITA03070
IF(<IFLGFY.EQ.1) WRITE(100UT,916) FOV,AJOB ITA03080
916 FORMAT(1H0,19X,9HF0V (DEG),5X,F7.3,1X,19H(ARBITRARY DEFAULT), ITA03090
* 4X,27HACQUISITION LEVEL (50 PCNT),3X,F7.3) ITA03100
IF(<IFLGMG.EQ.0) WRITE(100UT,917) AMAG,ZONE ITA03110
917 FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,24X,24HSEARCH ZONE (DEGREE ITA03120
*S**2),6X,F7.3) ITA03130
IF(<IFLGMG.EQ.1) WRITE(100UT,918) AMAG,ZONE ITA03140
918 FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,1X,19H(ARBITRARY DEFAULT), ITA03150
* 4X,24HSERACH ZONE (DEGREES**2),6X,F7.3) ITA03160
WRITE (100UT,920) ITA03170
920 FORMAT(1H0,52X,35HREQUIRES (TO DEFEAT DEVICE) AT MOST/ ITA03180
* 4X,14HFOR NO GREATER,6X,5HUNDER,8X,3HAND,7X,2HAT,4X, ITA03190
*35(1H-)) ITA03200
WRITE(100UT,919) ITA03210
919 FORMAT(8X,5HINPUT,11X,5HINPUT,7X,5HINPUT,5X,5HINPUT,3X, ITA03220
* 8HCOMPUTED,2X,11HCONTRAST OR,3X,8HCOMPUTED/6X, ITA03230
* 9HDETECTION,6X,11HAMB, ILLUM,2X,10HSKY/GROUND,2X, ITA03240
* 5H RANGE,2X,10HRESOLVABLE,1X,11HTEMP. DIFF.,2X,10HTOTAL PATH ITA03250
* /3X,5HPROB.,2X,9HTIME(SEC),3X,9H(FT CDL9),5X,5HRATIO,5X, ITA03260
* 4H(KM),3X,10HCYCLES, RC,2X,9HAT DEVICE,2X,13HTRANSMITTANCE, ITA03270
* 13X,8HCOMMENTS/3X,5(1H-),2X,9(1H-),2X,11(1H-),2X,10(1H-), ITA03280
* 1X,6(1H-),2X,10(1H-),2X,9(1H-),2X,13(1H-),2X,30(1H-) ITA03290
ILINE=23 ITA03300
C ***** BEGIN COMPUTATIONS FOV,RC,TBAR ITA03310
11 S=DIM/R ITA03320
IF (<FOV.LE.0.) FOV=.0001 ITA03330
TS=1.7*ZONE/FOV**2 ITA03340
IF(<ZONE.GT.9.0.AND.FOV.GT.5.) TS=(1.7*ZONE)/(5.0*FOV) ITA03350
IF (<TS.LT..5) TS=.5 ITA03360
CALL CYCLEX(PS,PB,AJOB,RC) ITA03370
IF(<AL.LE.0.) AL=1.E-7 ITA03380
ALPRNT=AL ITA03390
IF (<LSC.EQ.13) GOTO 13 ITA03400
TBAR=0. ITA03410
IF (<RC.LT.0.1) GOTO 14 ITA03420
IF(<(LSC.GT.5.AND.LSC.LT.10).OR.LSC.EQ.11.OR.LSC.EQ.14) GO TO 12 ITA03430
RC$=RC ITA03440
PINF=1.-EXP(-1.7*RC$/6.8) ITA03450
IF (<PINF.LE.0.) PINF=.0001 ITA03460
TBAR=0.5*TS*(2.-PINF)/PINF ITA03470
GO TO 14 ITA03480
12 TBAR=0.5*TS*(2.0-PS)/PS ITA03490
GO TO 14 ITA03500

```

```

C ***** DEVICE #13
13    NAL=1          ITA03510
      ICM=0          ITA03520
      PS=.99          ITA03530
      TBAR=1.8        ITA03540
      C=ACON          ITA03550
      TTOT=1          ITA03560
      TFNL=TTOT       ITA03570
      GO TO 22        ITA03580
C ***** COMPUTE C BASED ON AMBIENT ILLUMINATION. INTERPOLATE IF NECESSARY
14    IF (<S.LE.0.) S=.0001 ITA03600
      IF (<RC.LT.0.) RC=0. ITA03610
      RC=RC/S          ITA03620
      IF (<AMAG.LE.0.) AMAG=.0001 ITA03630
      IF(<LSC.EQ.1.OR.LSC.EQ.2) RC=RC/AMAG ITA03640
      IF(LSC.EQ.2) AL=0.7*AL ITA03650
C ***** CORRECT FOR FIELD OF VIEW
      IF(<LSC.LT.6.OR.LSC.EQ.7) GOTO 140 ITA03660
      IF(<(LSC.EQ.6).AND.<(MODE.EQ.2)) RC=RC/3 ITA03670
      IF(<(LSC.EQ.8).AND.<(MODE.EQ.1)) RC=RC*3 ITA03680
      IF(<(LSC.EQ.9).AND.<(MODE.EQ.1)) RC=RC*4 ITA03690
      IF(<(LSC.EQ.11).AND.<(MODE.EQ.1)) RC=RC*3 ITA03700
      IF(<(LSC.EQ.14).AND.<(MODE.EQ.1)) RC=RC*3 ITA03710
      IF(<(LSC.EQ.10).AND.<(MODE.EQ.1)) RC=RC*8 ITA03720
      IF(<(LSC.EQ.12).AND.<(MODE.EQ.2)) RC=RC/4 ITA03730
140   CONTINUE          ITA03740
      IF(<LSC.EQ.2) IAL=0 ITA03750
      IF(IAL.GT.0) GO TO 20 ITA03760
      AV=100.           ITA03770
      IF(AL.GT.100.) AL=100. ITA03780
      DO 15 I=1,9       ITA03790
      IF<(AL.GT.(0.999*AV)) GO TO 16 ITA03800
      AV=AV/10.         ITA03810
15    CONTINUE          ITA03820
      I=2              ITA03830
      IF(<LSC.GT.5) GOTO 160 ITA03840
      NAL=0            ITA03850
      ICM=0            ITA03860
      GO TO 22          ITA03870
C ***** CHECK IF INTERPOLATION NEEDED, IF NOT, GO TO 20
16    IF(<LSC.GT.5) GOTO 160 ITA03880
      IF<(AL.GT.(1.001*AV)) GO TO 17 ITA03890
160   IAL=I            ITA03900
      GO TO 20          ITA03910
C ***** INTERPOLATE
17    NAL1=I-1          ITA03920
      NAL2=I            ITA03930
      ICM1=0            ITA03940
      ICM2=0            ITA03950
      CALL INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA,AMRAL,
      *AMRGM)          ITA03960
      NAL=0            ITA03970
      IF(NAL1.EQ.0.AND.NAL2.EQ.0) GO TO 22 ITA03980
      NAL=NAL2          ITA03990
      IF(NAL1.GT.0) GO TO 18 ITA04000
      ICM=ICM2          ITA04010
      GO TO 21          ITA04020
18    IF(NAL2.GT.0) GO TO 19 ITA04030
      NAL=NAL1          ITA04040
      ICM=ICM1          ITA04050
      GO TO 21          ITA04060
19    IF<(ICM1.GT.0.AND.ICM2.GT.0) ICM=1 ITA04070
      IF<(ICM1.LT.0.AND.ICM2.LT.0) ICM=-1 ITA04080
      GO TO 21          ITA04090
C ***** NO INTERPOLATION NEEDED
20    NAL=IAL          ITA04100
      CALL CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM) ITA04110
21    IF<(ICM.EQ.-1) C=.99*C ITA04120
      CALL TREQ(ACON,SOC,C,LSC,TTOT) ITA04130
      IF<(ICM.EQ.1) TTOT=1 ITA04140
                                         ITA04150
                                         ITA04160
                                         ITA04170
                                         ITA04180
                                         ITA04190
                                         ITA04200

```

```

TFNL=TTOT
C ***** PRINT RESULTS
22 IF(NAL.GT.0) GO TO 23
TFNL=1
930 WRITE(100UT,930) PS,TBAR,ALPRNT,SOG,R,TFNL
FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,4X,6<1H->,7X,5<1H->,5X,
      *      F8.3,6X,2BHNOTE- AMBIENT ILLUM. OUTSIDE/91X,
      *      29HDEVICE OPERATIONAL LIMITS. NO/91X,
      *      21HOBSCURATION REQUIRED./1X)
ILINE=ILINE+3
GO TO 4
23 IF (TTOT.GE.1, AND, ICM.EQ.0) GOTO 24
IF<ICM.EQ.0> WRITE(100UT,931) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT
IF<ICM.EQ.1> WRITE(100UT,932) PS,TBAR,ALPRNT,SOG,R,RC,TTOT,C
IF<ICM.EQ.1> ILINE=ILINE+6
IF<ICM.EQ.-1> WRITE(100UT,933) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT
IF<ICM.EQ.-1> ILINE=ILINE+7
GO TO 4
24 WRITE(100UT,934) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT
FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,
      *      31HNOTE- CONTRAST (OR TEMP. DIFF.)/91X,
      *      31HREQUIRED WOULD EXCEED INTRINSIC/91X,
      *      31HCONTRAST (TEMP. DIFF.). NO OBS-/91X,
      *      17HCURART REQUIRED/1X)
ILINE=ILINE+3
GO TO 4
931 FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3)
932 FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,7X,5<1H->,5X,F8.3,6X,
      *      29HNOTE- DETECTION PROBABILITY /91X,
      *      30HREQUIRES CONTRAST (OR TEMP. /91X,
      *      29HDIFF.) AND RESOLVABLE CYCLES /91X,
      *      27HABOVE LIMIT FOR DEVICE. NO /91X,
      *      26HOBSCURANT REQUIRED. DEVICE/91X,
      *      18HUPPER LIMIT IS C= F8.3/1X)
933 FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,
      *      30HNOTE- INPUT DETECTION PROBAB- /91X,
      *      30HILITY REQUIRES CONTRAST (OR /91X,
      *      30HTEMP. DIFF.) BELOW THRESHOLD /91X,
      *      29HVALUES ASSUMED ARE 99 PERCENT/91X,
      *      29HF THRESHOLD. ADDITIONAL OBS-/91X,
      *      29HCURART WILL NOT DECREASE /91X,
      *      30HDETECTION PROBABILITY. /1X)
END

```

```

SUBROUTINE CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)      CIN00010
DIMENSION ALFA(9),BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)          CIN00020
COMMON /IOUNIT/IIN,IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCIN00030
DATA IPOS,INEG /1,-1/                                         CIN00040
C THIS ROUTINE COMPUTES THE REQUIRED CONTRAST AT THE DEVICE FOR A   CIN00050
GIVEN NUMBER OF RESOLVABLE CYCLES AT THE DEVICE. EQUATIONS           CIN00060
ARE INVERSIONS OF THOSE OF THE NV&EOL TARGET ACQUISITION            CIN00070
(FOR STATIC DETECTION) MODEL.                                         CIN00080
CIN00090
C INPUTS: LSC    = DEVICE NUMBER (1-14)                         CIN00100
C          RC     = RESOLVABLE CYCLES (DECIMAL)                      CIN00110
C          NAL    = AMBIENT ILLUMINATION CATEGORY (1-9)             CIN00120
CIN00130
C OUTPUTS: C     = CONTRAST OR TEMPERATURE DIFFERENCE(DEG K) REQUIRED CIN00140
C BY DEVICE FOR GIVEN RC.                                         CIN00150
C          ICM    = OPERATIONAL LIMITS FLAG (+1 EXCEEDS LIMIT, -1    CIN00160
C                   BELOW LIMIT, 0 WITHIN RANGE)                      CIN00170
C          NAL    = SET TO 0 IF TOO MUCH OR NOT ENOUGH AMBIENT ILLUM. CIN00180
CIN00190
CIFUN(RX,ALF,BET,GAM)=(BET+ALF*RX)/(GAM-RX)                     CIN00200
ICM=0                                         CIN00210
IF(NAL,EQ,0) GO TO 9020                                         CIN00220
C ***** BRANCH TO LSC                                         CIN00230
5   GO TO (10,10,30,40,50,60,60,80,90,100,1111,1200,9060,1400), LSC CIN00240
10  GO TO (110,120,130,140,150,160,170,9060,9060), NAL           CIN00250
C ***** DEVICE #1 NAL #1                                         CIN00260
110 IF(<2.74,LT,RC) GO TO 9030                                     CIN00270
IF(<2.133,LT,RC),AND,<(RC,LE,2.74)> GO TO 112                  CIN00280
IF(<2.26795,LT,RC),AND,<(RC,LE,2.133)> GO TO 113                CIN00290
IF<(RC,LE.,2.26795) GO TO 114                                    CIN00300
GO TO 9000                                         CIN00310
112 IEQ=1                                         CIN00320
GO TO 9050                                         CIN00330
113 IEQ=2                                         CIN00340
GO TO 9050                                         CIN00350
114 C=0,.015                                       CIN00360
GO TO 9015                                         CIN00370
C ***** DEVICE #1 NAL #2                                         CIN00380
120 IF(<2.74,LT,RC) GO TO 9030                                     CIN00390
IF(<2.133,LT,RC),AND,<(RC,LE,2.74)> GO TO 122                  CIN00400
IF(<0.0001,LE,RC),AND,<(RC,LE,2.133)> GO TO 123                CIN00410
IF<(RC,LT,0.0001) GO TO 124                                    CIN00420
GO TO 9000                                         CIN00430
122 IEQ=3                                         CIN00440
GO TO 9050                                         CIN00450
123 IEQ=4                                         CIN00460
GO TO 9050                                         CIN00470
124 C=0,.025                                       CIN00480
GO TO 9015                                         CIN00490
C ***** DEVICE #1 NAL #3                                         CIN00500
130 IF(<2.29,LT,RC) GO TO 131                                     CIN00510
IF(<0.49585,LE,RC),AND,<(RC,LE,2.29)> GO TO 132                CIN00520
IF<(RC,LT,0.49585) GO TO 133                                    CIN00530
GO TO 9000                                         CIN00540
131 C=0,.6324                                       CIN00550
GO TO 9035                                         CIN00560
132 IEQ=5                                         CIN00570
GO TO 9050                                         CIN00580
133 C=0,.030                                       CIN00590
GO TO 9015                                         CIN00600
C ***** DEVICE #1 NAL #4                                         CIN00610
140 IF(<1.5219,LT,RC) GO TO 141                                     CIN00620
IF(<0.313,LE,RC),AND,<(RC,LE,1.5219)> GO TO 142                CIN00630
IF<(RC,LT,0.313) GO TO 143                                    CIN00640
GO TO 9000                                         CIN00650
141 C=0,.70                                         CIN00660
GO TO 9035                                         CIN00670
142 IEQ=6                                         CIN00680
GO TO 9050                                         CIN00690
143 C=0.05                                         CIN00700

```

GO TO 9015	CIN00710
C ***** DEVICE #1 NAL #5	CIN00720
150 IF<1,1959,LT,RC> GO TO 9030	CIN00730
IF<(0,0,LE,RC),AND,<RC,LE,1,1959>> GO TO 152	CIN00740
IF<RC,LT,0,0> GO TO 153	CIN00750
GO TO 9000	CIN00760
152 IEQ=7	CIN00770
GO TO 9050	CIN00780
153 C=0.0841	CIN00790
GO TO 9015	CIN00800
C ***** DEVICE #1 NAL #6	CIN00810
160 IF<0,44767,LT,RC> GO TO 9030	CIN00820
IF<(0,0309,LE,RC),AND,<RC,LE,0,44767>> GO TO 162	CIN00830
IF<RC,LT,0,0309> GO TO 163	CIN00840
GO TO 9000	CIN00850
162 IEQ=8	CIN00860
GO TO 9050	CIN00870
163 C=0.18	CIN00880
GO TO 9015	CIN00890
C ***** DEVICE #1 NAL #7	CIN00900
170 IF<0,14,LT,RC> GO TO 9030	CIN00910
IF<(0,05,LE,RC),AND,<RC,LE,0,14>> GO TO 172	CIN00920
IF<RC,LT,0,05> GO TO 173	CIN00930
GO TO 9000	CIN00940
172 IEQ=9	CIN00950
GO TO 9050	CIN00960
173 C=0.50	CIN00970
GO TO 9015	CIN00980
C ***** DEVICE #3	CIN00990
360 GO TO <9060,9060,9060,340,350,360,370,380,390>, NAL	CIN01000
C ***** DEVICE #3 NAL #4	CIN01010
340 IF<0,7497,LE,RC,AND,RC,LT,2,941> GO TO 342	CIN01020
IF<(RC,GE,2,941)> GOTO 9075	CIN01030
IF<(RC,LT,0,7497)> GO TO 9040	CIN01040
GO TO 9000	CIN01050
342 IEQ=1	CIN01060
GO TO 9080	CIN01070
C ***** DEVICE #3 NAL #5	CIN01080
350 IF<(0,3988,LE,RC,AND,RC,LT,2,4350)> GO TO 352	CIN01090
IF<(RC,GE,2,4350)> GOTO 9075	CIN01100
IF<(RC,LT,0,3988)> GO TO 9040	CIN01110
GO TO 9000	CIN01120
352 IEQ=2	CIN01130
GO TO 9080	CIN01140
C ***** DEVICE #3 NAL #6	CIN01150
360 IF<(0,15965,LE,RC,AND,RC,LT,2,1060)> GO TO 362	CIN01160
IF<(RC,GE,2,106)> GOTO 9075	CIN01170
IF<(RC,LT,0,15965)> GO TO 9040	CIN01180
GO TO 9000	CIN01190
362 IEQ=3	CIN01200
GO TO 9080	CIN01210
C ***** DEVICE #3 NAL #7	CIN01220
370 IF<(0,05498,LE,RC,AND,RC,LT,2,0375)> GO TO 372	CIN01230
IF<(RC,GE,2,0375)> GOTO 9075	CIN01240
IF<(RC,LT,0,05498)> GO TO 9040	CIN01250
GO TO 9000	CIN01260
372 IEQ=4	CIN01270
GO TO 9080	CIN01280
C ***** DEVICE #3 NAL #8	CIN01290
380 IF<(0,442,LE,RC,AND,RC,LT,3,2190)> GO TO 382	CIN01300
IF<(RC,GE,3,219)> GOTO 9075	CIN01310
IF<(RC,LT,0,442),AND,<(RC,GT,0,>>> GO TO 383	CIN01320
IF<(RC,LE,0,)> GOTO 384	CIN01330
GO TO 9000	CIN01340
382 IEQ=5	CIN01350
GO TO 9080	CIN01360
383 C=0.57	CIN01370
GO TO 9015	CIN01380
384 C=0.33	CIN01390
GOTO 9015	CIN01400

C ***** DEVICE #3 NAL #9
 390 IF <0.172,LE,RC> GO TO 392
 IF<RC.LT.0.172,AND,RC.GT.0.> GO TO 393
 IF <RC,LE,0.> GOTO 394
 GO TO 9000

392 IEO=6
 GO TO 9080

393 C=0.57
 GO TO 9015

394 C=.33
 GOTU 9015

C ***** DEVICE #4 NAL #4
 40 GO TU <9060,9060,9060,440,450,460,470,480,490>, NAL

C ***** DEVICE #4 NAL #4
 440 IF <1.199,LE,RC,AND,RC,LT,5.2180> GO TO 442
 IF <RC,GE,5.218> GOTO 9075
 IF<RC.LT,1.199> GO TO 9040
 GO TO 9000

442 IEO=1
 GO TO 9080

C ***** DEVICE #4 NAL #5
 450 IF <0.6402,LE,RC,AND,RC,LT,4.2770> GO TO 452
 IF <RC,GE,4.277> GOTU 9075
 IF<RC.LT,0.6402> GO TO 9040
 GO TO 9000

452 IEO=2
 GO TO 9080

C ***** DEVICE #4 NAL #6
 460 IF <0.2449,LE,RC,AND,RC,LT,4.1240> GO TO 462
 IF <RC,GE,4.124> GOTO 9075
 IF<RC,LT,0.2449> GO TO 9040
 GO TO 9000

462 IEO=3
 GO TO 9080

C ***** DEVICE #4 NAL #7
 470 IF <0.08791,LE,RC,AND,RC,LT,3.4200> GO TO 472
 IF <RC,GE,3.420> GOTU 9075
 IF<RC,LT,0.08791> GO TO 9040
 GO TO 9000

472 IEO=4
 GO TO 9080

C ***** DEVICE #4 NAL #8
 480 IF <0.4394,LE,RC,AND,RC,LT,8.5970> GO TO 482
 IF <RC,GE,8.597> GOTU 9075
 IF<RC,LT,0.4394>,AND,(RC.GT.0.)> GO TO 483
 IF <RC,LE,0.> GOTO 484
 GO TO 9000

482 IEO=5
 GO TO 9080

483 C=0.33
 GO TO 9015

484 C=0.07
 GOTU 9015

C ***** DEVICE #1 NAL #9
 490 IF <0.1605,LE,RC> GO TO 492
 IF<RC,LT,0.1605,AND,RC.GT.0.> GO TO 493
 IF <RC,LE,0.> GOTO 494
 GO TO 9000

492 IEO=6
 GO TO 9080

493 C=0.33
 GO TO 9015

494 C=0.07
 GOTU 9015

C ***** DEVICE # 5
 50 GO TO <9060,9060,9060,540,550,560,570,580,590>, NAL

540 IF <0.9189,LE,RC,AND,RC,LT,2.8060> GO TO 542
 IF <RC,GE,2.806> GOTO 9075
 IF<RC,LT,0.9189> GO TO 9040
 GO TO 9000

CIN01410
 CIN01420
 CIN01430
 CIN01440
 CIN01450
 CIN01460
 CIN01470
 CIN01480
 CIN01490
 CIN01500
 CIN01510
 CIN01520
 CIN01530
 CIN01540
 CIN01550
 CIN01560
 CIN01570
 CIN01580
 CIN01590
 CIN01600
 CIN01610
 CIN01620
 CIN01630
 CIN01640
 CIN01650
 CIN01660
 CIN01670
 CIN01680
 CIN01690
 CIN01700
 CIN01710
 CIN01720
 CIN01730
 CIN01740
 CIN01750
 CIN01760
 CIN01770
 CIN01780
 CIN01790
 CIN01800
 CIN01810
 CIN01820
 CIN01830
 CIN01840
 CIN01850
 CIN01860
 CIN01870
 CIN01880
 CIN01890
 CIN01900
 CIN01910
 CIN01920
 CIN01930
 CIN01940
 CIN01950
 CIN01960
 CIN01970
 CIN01980
 CIN01990
 CIN02000
 CIN02010
 CIN02020
 CIN02030
 CIN02040
 CIN02050
 CIN02060
 CIN02070
 CIN02080
 CIN02090
 CIN02100

542	IEQ=1	CIN02110
	GO TO 9080	CIN02120
C ***** DEVICE #5	NAL #5	CIN02130
550	IF <0,59035,LE,RC,AND,RC,LT,2,7010> GO TO 552	CIN02140
	IF <RC,GE,2,7010> GOTO 9075	CIN02150
	IF<RC,LT,0,59035> GO TO 9040	CIN02160
	GO TO 9000	CIN02170
552	IEQ=2	CIN02180
	GO TO 9080	CIN02190
C ***** DEVICE #5	NAL #6	CIN02200
560	IF<0,2881,LE,RC,AND,RC,LT,2,4800> GO TO 562	CIN02210
	IF <RC,GE,2,4800> GOTO 9075	CIN02220
	IF<RC,LT,0,2881> GO TO 9040	CIN02230
	GO TO 9000	CIN02240
562	IEQ=3	CIN02250
	GO TO 9080	CIN02260
C ***** DEVICE #5	NAL #7	CIN02270
570	IF <0,12943,LE,RC,AND,RC,LT,1,6270> GO TO 572	CIN02280
	IF <RC,GE,1,6270> GOTO 9075	CIN02290
	IF<RC,LT,0,12943> GO TO 9040	CIN02300
	GO TO 9000	CIN02310
572	IEQ=4	CIN02320
	GO TO 9080	CIN02330
C ***** DEVICE #5	NAL #8	CIN02340
580	IF <0,4949,LE,RC,AND,RC,LT,2,6960> GO TO 582	CIN02350
	IF <RC,GE,2,6960> GOTO 9075	CIN02360
	IF<RC,LT,0,4949,AND,RC,GT,0,> GO TO 583	CIN02370
	IF <RC,LE,0,> GOTO 584	CIN02380
	GO TO 9000	CIN02390
582	IEQ=5	CIN02400
	GO TO 9080	CIN02410
583	C=0,33	CIN02420
	GO TO 9015	CIN02430
584	C=0,07	CIN02440
	GOTO 9015	CIN02450
C ***** DEVICE #5	NAL #9	CIN02460
590	IF <0,1894,LE,RC,AND,RC,LT,3,2898> GO TO 592	CIN02470
	IF <RC,GE,3,2898> GOTO 9075	CIN02480
	IF<RC,LT,0,1894,AND,RC,GT,0,> GO TO 593	CIN02490
	IF <RC,LE,0,> GOTO 594	CIN02500
	GO TO 9000	CIN02510
592	IEQ=6	CIN02520
	GO TO 9080	CIN02530
593	C=0,33	CIN02540
	GO TO 9015	CIN02550
594	C=0,07	CIN02560
	GOTO 9015	CIN02570
C ***** DEVICES #6 AND #7		CIN02580
60	IF<1,7934,LT,RC> GO TO 601	CIN02590
	IF<<0,92376,LT,RC>,AND,<RC,LE,1,7934>> GO TO 602	CIN02600
	IF<<0,11022,LE,RC>,AND,<RC,LE,0,92376>> GO TO 603	CIN02610
	IF<RC,LT,0,11022> GO TO 9070	CIN02620
	GO TO 9000	CIN02630
601	C=8,38	CIN02640
	GO TO 9035	CIN02650
602	C=<1,72363*RC-0,0392775>/<1,82560-RC>	CIN02660
	GO TO 9020	CIN02670
603	C=<0,09772*RC>/<1,-0,34779*RC>	CIN02680
	GO TO 9020	CIN02690
C ***** DEVICE #8		CIN02700
80	IF<4,702,LT,RC> GO TO 801	CIN02710
	IF<<0,996,LE,RC>,AND,<RC,LE,4,702>> GO TO 802	CIN02720
	IF<RC,LT,0,996> GO TO 803	CIN02730
	GO TO 9000	CIN02740
801	C=2,18	CIN02750
	GO TO 9035	CIN02760
802	C=<0,0298*RC>/<1,-0,199*RC>	CIN02770
	GO TO 9020	CIN02780
803	C=0,037	CIN02790
	GO TO 9015	CIN02800

```

C ***** DEVICE #9
90    IF(0.37181,LE,RC,AND,RC,LT,9.14027) GO TO 901
      IF (RC,GE,9.14027) GOTO 9075
      IF(RC,LT,0.37181) GO TO 9070
      GO TO 9000
901   C=<0.0289*RC)/(1.-0.1092*RC>
      GO TO 9020
C ***** DEVICE #10
100   IF(NAL,GT,2) GO TO 9060
      IF(4.227,LE,RC,AND,RC,LT,91.05956) GO TO 1010
      IF (RC,GE,91.05956) GOTO 9075
      C=0.01
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
C=C/1.63
      GO TO 9015
1010  C=<0.0303*RC)/(13.43-0.1473*RC>
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
C=C/1.63
      GO TO 9020
C ***** DEVICE # 11
1111  IF(0.5057,LE,RC,AND,RC,LT,7.73353) GO TO 1110
      IF (RC,GE,7.73353) GOTO 9075
      GO TO 9070
1110   C=<0.0207*RC)/(1.-0.1291*RC>
      GO TO 9020
C ***** DEVICE # 12
1200   IF(NAL,GT,2) GO TO 9060
      IF(3.48,LE,RC,AND,RC,LT,31.48113) GO TO 1210
      IF (RC,GE,31.41883) GOTO 9075
      C=0.01
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
C=C/1.63
      GO TO 9015
1210   C=<0.0206*RC)/(8.06-0.2559*RC>
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
C=C/1.63
      GO TO 9020
C ***** DEVICE # 14
1400   IF(0.9098,LE,RC,AND,RC,LT,4.57711) GO TO 1410
      IF (RC,GE,4.57711) GOTO 9075
      C=0.037
      GO TO 9015
1410   C=<0.0297567*RC)/(0.91287-0.1991449*RC>
      GO TO 9020
9000   WRITE(1000T,9010) LSC,NAL,RC
9010   FORMAT(5X,46HBAD PARAMETER PASSED TO SUBRTN CINV **** LSC= ,I3,
      *          6H NAL= ,I3,5H RC= ,F10.4)
      ICM=INEG
9020   RETURN
9030   C=0.80
9035   ICM=IPOS
      GO TO 9020
9040   C=0.02
      GO TO 9015
9050   C=CFUN(RC,ALFA(IEQ),BETA(IEQ),GAMA(IEQ))
      GO TO 9020
9060   NAL=0
      GO TO 9020
9070   C=0.0112
      GO TO 9015
9075   C=100.
      GOTO 9035
9080   L=LSC-2
      C=CFUN(RC,AMRAL(IEQ,L),0.,AMRGM(IEQ,L))
      GOTO 9020
      END

```

```

SUBROUTINE INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA,
*AMRAL,AMRGM)

INTERPOLATION ON THE SURFACE IN C, RC, AMBIENT ILLUMINATION SPACE.
THE USER INPUTS A VALUE OF RC, AND AN AMBIENT ILLUMINATION THAT
LIES BETWEEN ONE PAIR OF MODELED RC VS C CURVES AT TWO DISCRETELY
MODELED AMBIENT ILLUMINATIONS. THE TECHNIQUE IS COMPLICATED BY A
REQUIREMENT TO REPRODUCE ALMOST EXACTLY THE INTERPOLATED VALUE
FROM RC VS C OF THE NVL TARGET ACQUISITION MODEL. THUS, WHILE
RC IS INPUT IN THIS INVERSION MODEL, THE INTERPOLATION IS BETWEEN
RC VALUES AT CONSTANT C OVER TWO AMB. ILLUM. REGIONS. THE
THRESHOLDS OR OPERATIONAL LIMITS OF THE DEVICE ARE TREATED SEPAR-
ATELY. IN ALL, FOUR TYPES OF INTERPOLATION SITUATIONS ARE
UTILIZED.

DIMENSION IRGNS(21),INDEX(21),RLC(25),RCU(25),CU(25),CL(25),
*IEQ1(36),IEQ2(36),RLW(36),RUP(36),CLW(36),CUP(36),ALFAK(9),
*BETAK(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)
COMMON /IOUNIT/IOIN,IOCUT,IPHFLN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C** NUMBER OF C REGIONS FOR INTERPOLATION CONSIDERATION, DEV. 1-5
DATA IRGNS /4,4,3,3,2,2,1,1,1,2,1,1,1,1,2,1,1,1,1,2,1/
C** STARTING INDEX FOR EACH DEVICE, AMB. ILLUM. PAIR.
DATA INDEX /1,5,9,12,15,17,19,20,21,22,24,25,26,27,28,30,31,
*32,33,34,36/
C** LOWER LIGHT LEVEL THRESHOLDS OF RC.
DATA RLC /
*.0001,.26795,.49585,.313,0,.0309,.05,.7497,.3988,
*.15965,.055,.442,.1742,1.199,.6402,.2449,.0879,.4394,
*.1605,.9189,.5903,.288,.1294,.4949,.1894 /
C** UPPER LIGHT LEVEL THRESHOLDS OF RC
DATA RCU /
*.74,.74,.29,1.5219,1.1959,.44767,.14,.2941,.2435,
*.2106,.80375,.3219,.3219,.5218,.4277,.4124,.342,.8579,
*.597,.2806,.2701,.248,1.627,.2696,.32898 /
C** VALUE OF C AT LOWER THRESHOLDS.
DATA CL /
*.015,.025,.03,.05,.0841,.18,.50,.02,.02,.02,.02,
*.02,.57,.57,.02,.02,.02,.33,.33,.02 /
C** VALUE OF C AT UPPER THRESHOLD
DATA CU /
*.80,.80,.6324,.70,.80,.80,.100,.100,.100,.100,
*.100,.100,.16.88,100,.100,.100,.100,.100,.16.17,100 /
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR UPPER AMB. ILL.
DATA IEQ1 /2,2,1,1,4,4,3,3,5,5,0,6,6,0,7,7,8,8,1,2,3,4,4,5,1,2,3,
*4,4,5,1,2,3,4,4,5/
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR LOWER AMB ILL.
DATA IEQ2 /-1,4,4,3,-1,5,5,0,-1,6,6,-1,7,7,-1,8,-1,9,2,3,4,-1,5,
*6,2,3,4,-1,5,6,2,3,4,-1,5,6/
C** LOWEST RC VALUE IN EACH REGION.
DATA RLW /
*.26794,2.118,2.133,.26794,.4958,1.8439,2.29,.312,
*.312,1.5119,0,0,1.1758,0,.03089,.03089,.05,
*.3987,.15964,.054,.054,.442,.174,.64,.244,.087,
*.087,.4394,.1605,.59,.287,.129,.129,.4948,.1894 /
C** LARGEST RC VALUE IN EACH REGION.
DATA RUP /
*.67890,2.1335,2.1637,2.74,.61280,2.133,2.5764,2.74,.739,
*.29,.29,.84083,1.5219,1.5219,.712,1.1959,.3751,.4477,
*.2942,.2435,.2106,.9023,.3219,.3219,.5218,.4277,.4124,
*.1.0396,8.597,8.597,2.81,.271,.249,.9575,.2696,.329 /
C** LOWEST VALUE OF C OVER EACH REGION.
DATA CLW /
*.015,.025,.250,.270,.025,.030,.270,.6324,.030,
*.050,.6324,.050,.084107,.70,.0841,.180,.180,.500,
*.020,.020,.020,.570,.570,.020,.020,.020,.020,
*.020,.330,.330,.020,.020,.020,.020,.330,.330 /
C** LARGEST VALUE OF C OVER EACH REGION.
DATA CUP /

```

```

*.025 .250 .270 .800 .030 .270 .6324 .800 .050 / INT00710
*.6324 .700 .084107 .70 .800 .180 .800 .500 .800 / INT00720
*.100 .100 .100 .570 .100 .16.88 .100 .100 .100 .100 / INT00730
*.330 .100 .16.17 .100 .100 .100 .330 .100 .100 .100 / INT00740
C** SET LIGHT LEVELS, FRACTIONAL INTERPOLANT. INT00750
    ICM1=0
    ICM2=0
    IF (<NAL1.EQ.NAL2) GOTO 10 INT00760
    IF (<LSC.GT.5) GOTO 10 INT00780
    AL1=10.**(<3-NAL1) INT00800
    AL2=10.**(<3-NAL2) INT00810
    FACTR=(AL-AL2)/(AL1-AL2) INT00820
    DFAC=1.-FACTR INT00830
    GOTO <(100,100,200,200),LSC INT00840
10    CALL CINV(LSC,RC,NAL1,C,ICM1,ALFA,BETA,GAMA,AMRAL,AMRGM) INT00850
20    NAL2=0 INT00860
    RETURN INT00870
30    CALL CINV(LSC,RC,NAL2,C,ICM2,ALFA,BETA,GAMA,AMRAL,AMRGM) INT00880
40    NAL1=0 INT00890
    RETURN INT00900
C** DEVICES 1,2 INT00910
100   IF (<NAL2.LE.7) GOTO 105 INT00920
    IF (<NAL1.EQ.7.AND.FACTR.GT.0.75) GOTO 10 INT00930
    NAL1=0 INT00940
    GOTO 20 INT00950
105   NRG=IRGNS(NAL1) INT00960
    IDX=INDEX(NAL1) INT00970
    LIM1=NAL1 INT00980
    LIM2=NAL2 INT00990
    GOTO 500 INT01000
C** DEVICES 3,4,5 INT01010
200   IF (<NAL1.GE.4) GOTO 205 INT01020
    IF (<NAL2.EQ.4.AND.FACTR.LT.0.3) GOTO 30 INT01030
    NAL2=0 INT01040
    GOTO 40 INT01050
205   IS=6+<(LSC-3)*5+<(NAL1-3) INT01060
    NRG=IRGNS(IS) INT01070
    IDX=INDEX(IS) INT01080
    LIM1=7+<(LSC-3)*6+<(NAL1-3) INT01090
    LIM2=LIM1+1 INT01100
    BVAL=0, INT01110
    IDY=LSC-2 INT01120
C** FIRST, CHECK LIMIT. INT01130
500   IF (<RC.GE.RCL(LIM1).AND.RC.GE.RCL(LIM2)) GOTO 510 INT01140
    CALL CASE3(FACTR,CNOT,RCN,RCL(LIM1),RCL(LIM2),CL(LIM1),CL(LIM2)) INT01150
    IF (<RC.GT.RCN) GOTO 520 INT01160
    CV=CNOT INT01170
    ICM1=-1 INT01180
    ICM2=-1 INT01190
    GOTO 800 INT01200
510   IF (<RC.LE.RCU(LIM1).AND.RC.LE.RCU(LIM2)) GOTO 520 INT01210
    CALL CASE3(FACTR,CNOT,RCN,RCU(LIM1),RCU(LIM2),CU(LIM1),CU(LIM2)) INT01220
    IF (<RC.LT.RCN) GOTO 520 INT01230
    CV=CNOT INT01240
    ICM1=1 INT01250
    ICM2=1 INT01260
    GOTO 800 INT01270
C** RC IS IN BOUNDS OF OPERATIONAL LIMITS, BUT NOT NECESSARILLY INT01280
C BETWEEN CURVES. TEST SUB-REGIONS IN TURN. INT01290
520   DO 600 I=1,NRG INT01300
    J=IDX+I-1 INT01310
    IF (<RC.LT.RLW(J).OR.RC.GT.RUP(J)) GOTO 600 INT01320
C** RC IS IN SUB-REGION. NOW TEST C INT01330
    CV=0 INT01340
    CR1=0 INT01350
    CR2=0 INT01360
    IQ1=IEQ1(J) INT01370
    IQ2=IEQ2(J) INT01380
    IF (<IQ1.GT.0) GOTO 530 INT01390
    RCN=RCU(LIM1) INT01400

```

```

CNOT=CU(LIM1) INT01410
IF (IQ1.LT.0) RCN=RCL(LIM1) INT01420
IF (IQ1.LT.0) CNOT=CL(LIM1) INT01430
IF (LSC.LE.2) CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ2),INT01440
*BETAC(IQ2),GAMA(IQ2)) INT01450
IF (LSC.GT.2)CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,INT01460
*AMRAL(IQ2,IDX),BVAL,AMRGMC(IQ2,IDX)) INT01470
IF (IFLG.NE.0) GOTO 600 INT01480
IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600 INT01490
IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600 INT01500
IF (RCN.EQ.RCU(LIM1)) ICM1=1 INT01510
IF (RCN.EQ.RCL(LIM1)) ICM1=-1 INT01520
GOTO 800 INT01530
530 IF (IQ2.GT.0) GOTO 540 INT01540
RCN=RCU(LIM2) INT01550
CNOT=CU(LIM2) INT01560
IF (IQ2.LT.0) RCN=RCL(LIM2) INT01570
IF (IQ2.LT.0) CNOT=CL(LIM2) INT01580
IF (LSC.LE.2) CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ1),INT01590
*BETAC(IQ1),GAMA(IQ1)) INT01600
IF (LSC.GT.2)CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,INT01610
*AMRAL(IQ1,IDX),BVAL,AMRGMC(IQ1,IDX)) INT01620
IF (IFLG.NE.0) GOTO 600 INT01630
IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600 INT01640
IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600 INT01650
IF (RCN.EQ.RCU(LIM2)) ICM2=1 INT01660
IF (RCN.EQ.RCL(LIM2)) ICM2=-1 INT01670
GOTO 800 INT01680
540 IF(LSC.LE.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,ALFA(IQ1),ALFA(IQ2),INT01690
*BETAC(IQ1),BETAC(IQ2),GAMA(IQ1),GAMA(IQ2)) INT01700
IF(LSC.GT.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,AMRAL(IQ1,IDX),INT01710
*AMRAL(IQ2,IDX),BVAL,BVAL,AMRGMC(IQ1,IDX),AMRGMC(IQ2,IDX)) INT01720
IF (IFLG.GE.2) GOTO 600 INT01730
CV=CR1 INT01740
IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800 INT01750
IF (IFLG.EQ.1) GOTO 600 INT01760
CV=CR2 INT01770
IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800 INT01780
600 CONTINUE INT01790
C** NO VALUES MET INTERPOLATION CRITERIA. INT01800
C** FINAL TEST IS IN THE LIMITING REGIONS INT01810
640 CONTINUE INT01820
WRITE (1000,LSC,AL1,AL2) INT01830
1000 FORMAT(1X,40H*** IN ITAM INTAL ROUTINE, DEVICE LSC = ,I2, INT01840
*/1X,48H*** COULD NOT INTERPOLATE BETWEEN AMBIENT ILLUM.,F11.6, INT01850
*5H AND ,F11.6/1X,39H*** UPPER AMBIENT ILLUM. VALUE ASSUMED.) INT01860
GOTO 10 INT01870
800 C=CV INT01880
RETURN INT01890
END INT01900

```

```

SUBROUTINE CASE1(FACTR,RC,CR1,CR2,IFLG,ALP1,ALP2,BET1,BET2,
*GAM1,GAM2)                               CSA00010
                                             CSA00020
                                             CSA00030
                                             CSA00040
                                             CSA00050
                                             CSA00060
                                             CSA00070
                                             CSA00080
                                             CSA00090
                                             CSA00100
                                             CSA00110
                                             CSA00120
                                             CSA00130
                                             CSA00140
                                             CSA00150
                                             CSA00160
                                             CSA00170
                                             CSA00180
                                             CSA00190
                                             CSA00200
                                             CSA00210
                                             CSA00220
                                             CSA00230
                                             CSA00240
                                             CSA00250
                                             CSA00260
                                             CSA00270
                                             CSA00280
                                             CSA00290
                                             CSA00300

C FINDS CONTRAST C INTERPOLATING ON RC VS C. RETURNS AT MOST TWO
C ROOTS CR1, CR2. IFLG=1 IF ROOTS ARE IDENTICAL, IFLG=2 IF ROOTS
C ARE COMPLEX.
C
DFAC=1.-FACTR
A1=RC-GAM1*FACTR-GAM2*DFAC
A2=RC*(ALP1+ALP2)+(BET1-ALP2*GAM1)*FACTR+(BET2-ALP1*GAM2)*DFAC
A3=ALP1*ALP2*RC+ALP2*BET1*FACTR+ALP1*BET2*DFAC
IFLG=2
IF (A1.LT.-1.E-10.OR.A1.GT.1.E-10) GOTO 10
IF (A2.EQ.0.) RETURN
IFLG=1
CR1=-A3/A2
CR2=CR1
RETURN
10 DISCR=A2*A2-4.*A1*A3
IF (DISCR.LT.0.) RETURN
IF (DISCR.EQ.0.) IFLG=1
IF (DISCR.GT.0.) IFLG=0
VA=SQRT(DSCR)/(2.*A1)
VB=-A2/(2.*A1)
SG=-1
IF (A1.LT.0.) SG=1.
CR1=VB+SG*VA
CR2=VB-SG*VA
RETURN
END

```

```

SUBROUTINE CASE2(FACTR,RC,C,IFLG,CP,RNOT,CNOT,ALP,BET,GAM)      CSB00010
C FINDS INTERPOLATED CONTRAST C FOR CASE OF SINGLE FIXED RC AT    CSB00020
C SOME LIMIT.                                                       CSB00030
C                                                               CSB00040
C                                                               CSB00050
C                                                               CSB00060
C                                                               CSB00070
C                                                               CSB00080
C                                                               CSB00090
C                                                               CSB00100
C                                                               CSB00110
C                                                               CSB00120
C                                                               CSB00130
C                                                               CSB00140
C                                                               CSB00150
C                                                               CSB00160
C                                                               CSB00170
C                                                               CSB00180
C                                                               CSB00190
C                                                               CSB00200
C                                                               CSB00210
C                                                               CSB00220
C                                                               CSB00230
C                                                               CSB00240
C                                                               CSB00250
C                                                               CSB00260
C                                                               CSB00270
C                                                               CSB00280
C                                                               CSB00290
C                                                               CSB00300
C                                                               CSB00310
C                                                               CSB00320
C                                                               CSB00330

IFLG=1
IF (FACTR.NE.1.) GOTO 10
IF (RC.NE.RNOT) RETURN
GOTO 90
10 IF (RC.EQ.RNOT) GOTO 25
DIFF=RC-RNOT*FACTR
IF (DIFF.NE.0.) GOTO 20
IF (GAM.EQ.0.) RETURN
CP=BET/GAM
GOTO 50
20 VA=BET*(1.-FACTR)+ALP*DIFF
VB=GAM*(1.-FACTR)-DIFF
IF (VB.EQ.0.) GOTO 90
CP=VA/VB
GOTO 50
25 IF (GAM.EQ.RC) RETURN
CP=(ALP*RC+BET)/(GAM-RC)
GOTO 60
50 RP=(GAM*CP-BET)/(ALP+CP)
C=CNOT+(CP-CNOT)*(RC-RNOT)/(RP-RNOT)
GOTO 100
60 C=CNOT*FACTR+CP*(1.-FACTR)
GOTO 100
90 C=CNOT
CP=CNOT
100 IFLG=0
RETURN
END

```

```
SUBROUTINE CASE3(FACTR,C,RCN,RNOT1,RNOT2,CNOT1,CNOT2)      CSC00010
C FINDS INTERPOLATED C FOR CASE OF BOTH RC VALUES FIXED AT    CSC00020
C LIMITS.                                                       CSC00030
                                                               CSC00040
                                                               CSC00050
                                                               CSC00060
                                                               CSC00070
                                                               CSC00080
                                                               CSC00090
10    RCN=RNOT1*FACTR+RNOT2*(1.-FACTR)                         CSC00100
      IF (< RNOT1, EQ, RNOT2 ) GOTO 10
      C=CNOT1+(CNOT2-CNOT1)*(RCN-RNOT1)/(RNOT2-RNOT1)
      RETURN
      C=CNOT1*FACTR+CNOT2*(1.-FACTR)
      RETURN
      END
```

```

SUBROUTINE CYCLE(PS,PB,AJOB,RC) CYC00010
C THIS ROUTINE COMPUTES THE NORMALIZED NUMBER OF RESOLVABLE CYCLES CYC00020
C FOR GIVEN CUMULATIVE NORMAL DISTRIBUTION OF PROBABILITY OF CYC00030
C DETECTION. CYC00040
C INPUTS: PS = CUMULATIVE PROBABILITY OF DETECTION. CYC00050
C PB = STANDARD DEVIATION FOR PS (.632) CYC00060
C AJOB = MEAN RESOLVABLE CYCLES FOR 50 PERCENT PROBABILITY CYC00070
C OF DETECTION. CYC00080
C OUTPUTS: RC = RESOLVABLE CYCLES REQUIRED FOR PS CYC00090
C SUBROUTINE: CYCLE CALLED BY ITAM. CALLS NONE. CYC00100
C
COMMON /IQUUNIT/IQIN,IQOUT,IPHFUN,LQUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCYC00140
DATA A0,A1,A2 /2.515517,0.802853,0.010328/ CYC00150
DATA B1,B2,B3 /1.432788,0.189269,0.001308/ CYC00160
ICASE=0 CYC00170
IF((0.0,LE,PS),AND,(PS,LE,0.0000003)) GO TO 10 CYC00180
IF((.9999997,LE,PS),AND,(PS,LE,1.)) GO TO 20 CYC00190
IF((.0000003,LT,PS),AND,(PS,LE,0.5)) GO TO 30 CYC00210
IF((.5,LT,PS),AND,(PS,LT,0.9999997)) GO TO 40 CYC00220
C ***** ERROR ROUTINE CYC00230
      WRITE(IQOUT,900) PS CYC00240
      RETURN CYC00250
C ***** SPECIAL CASE I CYC00260
10   X=-5 CYC00270
      IF(PS,EQ,0.) X=-45 CYC00280
      GO TO 60 CYC00290
C ***** SPECIAL CASE II CYC00300
20   X=5 CYC00310
      IF(PS,EQ,1.) X=45 CYC00320
      GO TO 60 CYC00330
C ***** CASE I CYC00340
30   XX=SQRT ALOG(1/(PS*PS)) CYC00350
      ICASE=1 CYC00360
      GO TO 50 CYC00370
C ***** CASE II CYC00380
40   XX=SQRT ALOG(1/(1.-PS)**2)) CYC00390
50   X=XX-(A0+A1*XX+A2*(XX*XX))/(1.+B1*XX+B2*(XX*XX)+B3*(XX*XX*XX)) CYC00400
60   RC=AJOB+AJOB*PB*X CYC00410
      RETURN CYC00420
900  FORMAT(2X,39HERROR - INCORRECT PS VALUE PASSED PS = ,F10.4) CYC00430
      END CYC00440

```

```

SUBROUTINE TREQ(ACON,SOG,C,LSC,TRQ) TRQ00010
C THIS ROUTINE COMPUTES THE TRANSMISSION REQUIRED TO MATCH GIVEN TRQ00020
C TARGET AND DETECTOR CONTRASTS, WITH MODIFICATION BY SKY/ TRQ00030
C GROUND RATIO FOR NON-THERMAL DEVICES. TRQ00040
C INPUTS: ACON = INTRINSIC CONTRAST OF TARGET, OR TEMP. DIFF. TRQ00050
C          (DEG K) FOR THERMAL. TRQ00060
C          SOG = SKY/GROUND RATIO. TRQ00070
C          C = CONTRAST OR TEMP. DIFF. SEEN AT DEVICE TRQ00080
C          LSC = DEVICE NUMBER. TRQ00090
C OUTPUTS: TRQ = REQUIRED TRANSMISSION TO REDUCE ACON TO C. TRQ00100
C SUBROUTINE: TREQ CALLED BY ITAM. CALLS NONE. TRQ00110
C
C COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUTRQ00120
C IF((LSC.LT.1),OR,(LSC.GT.14)) GO TO 30 TRQ00130
C IF(LSC.GT.5) GO TO 20 TRQ00140
C **** NON-THERMAL TRQ00150
10  IF (SOG.EQ.0, OR, C.EQ.0,) GOTO 45 TRQ00160
    TRQ=1./((1.+((ACON-C)/(C+SOG))) TRQ00170
    GOTO 50 TRQ00180
20  IF((LSC.EQ.10),OR,(LSC.EQ.12)) GO TO 10 TRQ00190
    IF(LSC.EQ.13) GO TO 30 TRQ00200
C **** THERMAL TRQ00210
    IF (ACON.EQ.0,) GOTO 25 TRQ00220
    TRQ=C/ACON TRQ00230
    GOTO 50 TRQ00240
25  TRQ=1. TRQ00250
    GOTO 50 TRQ00260
C **** ERROR ROUTINE TRQ00270
30  WRITE(C100OUT,100) LSC TRQ00280
    TRQ=1. TRQ00290
100 FORMAT(5X,19HINPUT ERROR LSC = ,I2) TRQ00300
    RETURN TRQ00310
45  TRQ=0. TRQ00320
50  IF (TRQ.LT.0.) TRQ=0. TRQ00330
    IF (TRQ.GT.1.) TRQ=1. TRQ00340
    RETURN TRQ00350
    END TRQ00360
    TRQ00370
    TRQ00380
    TRQ00390

```

```

SUBROUTINE FCLLOUD(LAMBDA,TRANS,IERR)
REAL L,M,N,L0,M0,N0,L0S,M0S,L0M,L0N,N0M,M0L,M0N
DIMENSION IALPH(10)
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRLU,NCLIMT,KSTOR,NPLOTU
COMMON /FGEOM/XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS
COMMON /OPT/ INDEXP,ETA,KAPPA,W0,THETAO,PHIO,LD,TAUBAR,
+RHO,LB0,TMPA,TMPC
COMMON /GEOMET/FTS(15),IGEOSW
DATA IZERO/0/
DATA IALPH/2HCP ,2HSP ,2HAX ,2HCL ,2HAT ,2HBK ,
+2HSA ,2HLU ,2HGD /
ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
*****FCL 00130
*****FCL 00140
*****FCL 00150
*****FCL 00160
*****FCL 00170
*****FCL 00180
*****FCL 00190
*****FCL 00200
*****FCL 00210
*****FCL 00220
*****FCL 00230
*****FCL 00240
*****FCL 00250
*****FCL 00260
*****FCL 00270
*****FCL 00280
*****FCL 00290
*****FCL 00300
*****FCL 00310
*****FCL 00320
*****FCL 00330
*****FCL 00340
*****FCL 00350
*****FCL 00360
*****FCL 00370
*****FCL 00380
*****FCL 00390
*****FCL 00400
*****FCL 00410
*****FCL 00420
*****FCL 00430
*****FCL 00440
*****FCL 00450
*****FCL 00460
*****FCL 00470
*****FCL 00480
*****FCL 00490
*****FCL 00500
*****FCL 00510
*****FCL 00520
*****FCL 00530
*****FCL 00540
*****FCL 00550
*****FCL 00560
*****FCL 00570
*****FCL 00580
*****FCL 00590
*****FCL 00600
*****FCL 00610
*****FCL 00620
*****FCL 00630
*****FCL 00640
*****FCL 00650
*****FCL 00660
*****FCL 00670
*****FCL 00680
*****FCL 00690

```

SUBROUTINE FCLLOUD COMPUTES THE FOLLOWING QUANTITIES FOR A FINITE ELLIPSOIDAL CLOUD:

S = LENGTH OF OPTICAL PATH IN CLOUD
TRANS = BEAM TRANSMITTANCE THROUGH CLOUD
LF = PATH RADIANCE
TC = CONTRAST TRANSMITTANCE.

INPUT OF COMPUTATIONAL PARAMETERS TAKES PLACE THROUGH AN ORDER INDEPENDENT READ OF A GROUP OF RECORDS IDENTIFIED BY THE FOLLOWING MNEMONICS (EACH RECORD IS FORMAT (A4,1X,5(E10.4,1X))) :

MNEUMONIC	VARIABLES READ	DESCRIPTION
CPOS	XC,YC,ZC	CLOUD CENTER POSITION
RPOS	XR,YR,ZR	RECEIVER POSITION
SPOS	XS,YS,ZS	SOURCE POSITION
AXES	AE,BE,CE	SEMI-AXES OF CLOUD ELLIPSOID
CLDS	INDEXP,ETA,KAPPA,W0,TMPC	CLOUD AEROSOL PARAMETERS
ATMO	TAUBAR,TMPA	ATMOSPHERIC PARAMETERS
BKGR	RHO,LB0	BACKGROUND PARAMETERS
SANG	THETAO,PHIO	SOLAR ANGLES
LUND	LD	LUNAR DAY
GO		TERMINATES READ

** NOTE : THE GO SENTINEL CARD MUST BE THE LAST CARD READ

THIS INPUT DATA IS STORED FOR LATER USE IN COMMON BLOCKS /FGEOM/ AND /OPT/.

/FGEOM/ INPUT PARAMETERS:

(XC,YC,ZC) = CENTER OF ELLIPSOID
(AE,BE,CE) = SEMI-AXES OF ELLIPSOID
(XR,YR,ZR) = COORDINATES OF RECEIVER LOCATION
(XS,YS,ZS) = COORDINATES OF SOURCE LOCATION.

/OPT/ INPUT PARAMETERS:

INDEXP = PHASE FUNCTION IDENTIFIER

- =0, USER SUPPLIED
- =1, MARITIME ARCTIC, VIS=0.1 TO 2.0 KM
- =2, MARITIME POLAR, VIS=0.2 KM
- =3, MARITIME POLAR, VIS=0.2 KM
- =4, CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM
- =5, WHITE PHOSPHORUS
- =6, HEXACHLOROETHANE
- =7, FOG OIL
- =8, DUST (MODERATE AEROSOL LOADING)
- =9, DUST (HEAVY AEROSOL LOADING)
- =10, MARITIME MODEL B, VIS=5KM, RH=95%
- =11, MARITIME MODEL B, VIS=10KM,RH=90%

=12, MARITIME MODEL B, VIS=50KM, RH=50% FCL00700
 ETA = FORWARD SCATTERING PARAMETER FCL00710
 ETA MAY ALSO BE CALCULATED BY DEFAULT (I.E. INPUT ETA = 0.0).
 IN THIS CASE INDEXP SHOULD BE THE NEGATIVE OF THE PHASE
 FUNCTION FOR WHICH ETA IS DESIRED.
 KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1) FCL00720
 W0 = SINGLE SCATTERING ALBEDO FCL00730
 (<THETA0,PHI0>) = SOLAR ANGLES (DEGREES) FCL00740
 LD = LUNAR DAY (INTEGER BETWEEN 0 AND 28) FCL00750
 TAUBAR = ATMOSPHERIC OPTICAL THICKNESS ABOVE CLOUD FCL00760
 RHO = BACKGROUND SURFACE ALBEDO FCL00770
 LBO = BACKGROUND RADIANCE FCL00780
 TMFA = TEMPERATURE OF ATMOSPHERE (DEG.C) FCL00790
 TMPC = TEMPERATURE OF CLOUD (DEG.C) FCL00800
 FCL00810
 ALL LENGTH UNITS ARE KM. PROGRAM FLOW IS CONTROLLED BY THE
 VARIABLES TMFA, INDEXP, AND, IMPLICITLY, BY ISW.
 IF TMFA >= -99.0 THIS SPECIFIES FCL00820
 A THERMAL COMPUTATION, WHICH IS PERFORMED IN SUBROUTINE THRMC1:
 IF TMFA < -99.0 THEN A SCATTERING COMPUTATION IS PERFORMED,
 IF INDEXP < 0, THIS IS A MULTIPLE SCATTERING COMPUTATION, WHICH
 IS DONE IN SUBROUTINE MSCLD; IF ETA HAS BEEN INPUT AS ZERO, THEN
 INDEXP SHOULD BE THE NEGATIVE OF THE PHASE FUNCTION IDENTIFIER,
 SO THAT ETA WILL BE FOUND FROM THE PROPER PHASE FUNCTION.
 IF INDEXP > 0 A SINGLE SCATTERING
 COMPUTATION IS CARRIED OUT IN SUBROUTINE SSCLD. IN THIS CASE FCL00890
 INDEXP ALSO SPECIFIES THE PHASE FUNCTION TO BE USED, WITH
 INDEXP = I SELECTING THE I-TH PHASE FUNCTION. THE VALUE OF FCL00900
 ISW OCCURRING IN THE SUBROUTINE PARAMETER LIST INDICATES WHETHER FCL00910
 CERTAIN PARAMETERS ARE THE SAME AS IN THE PREVIOUS CALL TO FCLOUD,
 AS FOLLOWS. FCL00920
 FCL00930
 FCL00940
 FCL00950
 ISW IS SET TO 0 WHEN THE FOLLOWING CONDITIONS ARE ENCOUNTERED :
 A) ALL 9 DATA CARDS ARE READ FCL00960
 B) ANY ONE OF CARDS 1-4 AND ANY ONE OF CARDS 5-9
 (AS LISTED IN THE ORDER ABOVE) ARE READ FCL00970
 FCL00980
 FCL00990
 ISW IS SET TO 2 IF NONE OF CARDS 1-4 IS READ.
 ISW IS SET TO 1 IF NONE OF CARDS 5-9 IS READ.
 ISW DEFAULTS TO 2 IF NOTHING IS READ (OTHER THAN THE GO CARD).
 FCL01000
 FCL01010
 FCL01020
 FCL01030
 FCL01040
 FCL01050
 ISW = 2 => PARAMETERS IN COMMON /FGEOM/ ARE THE SAME AS IN
 PREVIOUS CALL; SKIP PRELIMINARY GEOMETRICAL
 CALCULATIONS FCL01060
 FCL01070
 ISW = 1 => PARAMETERS IN COMMON /OPT/ ARE THE SAME AS IN
 PREVIOUS CALL; SKIP COMPUTATIONS INVOLVING
 ONLY THESE PARAMETERS FCL01080
 FCL01090
 ISW = 0 => NEW PARAMETERS IN BOTH /FGEOM/ AND /OPT/;
 NO CALCULATIONS ARE SKIPPED IN THIS CASE. FCL01100
 FCL01110
 FCL01120
 FCL01130
 FCL01140
 SUBROUTINES CALLED FROM FCLOUD:
 THRMC1 FOR THERMAL CALCULATION OF PATH RADIANCE FCL01150
 MSCLD FOR MULTIPLE SCATTERING CALCULATION OF
 PATH RADIANCE FCL01160
 SSCLD FOR SINGLE SCATTERING CALCULATION OF PATH
 RADIANCE FCL01170
 ILLUM COMPUTES THE EXTRATERRESTRIAL SOLAR OR
 LUNAR IRRADIANCE. NOT NEEDED FOR THERMAL
 CALCULATIONS FCL01180
 FCL01190
 PFN RETURNS A PHASE FUNCTION VALUE FOR USE BY
 SSCLD. FCL01200
 FCL01210
 FCL01220
 FCL01230
 FCL01240
 FCL01250
 GROUND RADIANCE CAN BE SPECIFIED IN ONE OF TWO WAYS:
 1.) BY GIVING LBO A NONZERO VALUE, OR FCL01260
 2.) AS REFLECTED SOLAR IRRADIANCE, USING THE VALUE OF
 RHO INPUT TO THE PROGRAM. FCL01270
 FCL01280
 FCL01290
 FCL01300
 THE VALUE OF LD SPECIFIES WHETHER IT IS DAY OR NIGHT, I. E.
 WHETHER SOLAR OR LUNAR EXTRATERRESTRIAL IRRADIANCE IS TO BE
 USED. IF LD = 0 THEN SOLAR IRRADIANCE IS CALCULATED, AND IF FCL01310
 FCL01320
 FCL01330

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

4 - 6

41

21021*

```

C   1 < LD < 28 THE IRRADIANCE VALUE IS FOR LUNAR DAY LD.          FCL01340
C ***** FCL01350
C ***** FCL01360
C ***** FCL01370
C ***** FCL01380
C ***** FCL01390
C
C   INITIALIZATION OF INPUT DATA.
C
C   IF (IZERO.NE.0) WRITE (IOOUT,99)                                     FCL01400
99   FORMAT(1H1)                                                       FCL01410
    IF (IZERO.NE.0) GO TO 477                                         FCL01420
    XC=0.0                                                               FCL01430
    YC=0.0                                                               FCL01440
    ZC=0.0                                                               FCL01450
    XR=0.0                                                               FCL01460
    YR=0.0                                                               FCL01470
    ZR=0.0                                                               FCL01480
    AE=0.0                                                               FCL01490
    BE=0.0                                                               FCL01500
    CE=0.0                                                               FCL01510
    INDEXP=0                                                               FCL01520
    ETA=0.0                                                               FCL01530
    KAPPA=0.0                                                               FCL01540
    W0=0.0                                                               FCL01550
    TMPC=0.0                                                               FCL01560
    TAUBAR=0.0                                                               FCL01570
    TMPA=0.0                                                               FCL01580
    RH0=0.0                                                               FCL01590
    LB0=0.0                                                               FCL01600
    THETA0=0.0                                                               FCL01610
    PHI0=0.0                                                               FCL01620
    LD=0                                                               FCL01630
    ISW=0                                                               FCL01640
    IZER0=1                                                               FCL01650
477   CONTINUE
    IFLG=2                                                               FCL01660
    IFLO=1                                                               FCL01670
    DO 360 K=1,10
      READ(10IN,334)IA,IA2,R1,R2,R3,R4,R5
334   FORMAT(2A2,1X,5(E10.4,1X))
      DO 333 I=1,11
        IF (IA.NE.IALPH(I)) GO TO 333
        IND=I
        IF (IND.EQ.10) GO TO 361
333   CONTINUE
        IF (K.EQ.10.AND.IND.NE.10) GO TO 358
        IF (IND.LT.5)IFLG=0
        IF (IND.GE.5.AND.IND.LE.9)IFLO=0
        IF (IND.EQ.11) GO TO 355
        IF (IND.LT.5) GO TO (341,342,343,344),IND
        IND4=IND-4
        GO TO (345,346,347,348,349),IND4
341   XC=R1                                                               FCL01760
      YC=R2                                                               FCL01770
      ZC=R3                                                               FCL01780
      GO TO 360
342   XR=R1                                                               FCL01790
      YR=R2                                                               FCL01800
      ZR=R3                                                               FCL01810
      GO TO 360
343   XS=R1                                                               FCL01820
      YS=R2                                                               FCL01830
      ZS=R3                                                               FCL01840
      GO TO 360
344   AE=R1                                                               FCL01850
      BE=R2                                                               FCL01860
      CE=R3                                                               FCL01870
      GO TO 360
345   INDEXP=IFIX(R1)                                                 FCL01880

```

```

IF (R2.GT..001) ETA=R2 FCL 02020
KAPPA=R3 FCL 02030
W0=R4 FCL 02040
TMPC=R5 FCL 02050
GO TO 360 FCL 02060
346 TAUBAR=R1 FCL 02070
TMPA=R2 FCL 02080
GO TO 360 FCL 02090
347 RH0=R1 FCL 02100
LB0=R2 FCL 02110
GO TO 360 FCL 02120
348 THETA0=R1 FCL 02130
PHI0=R2 FCL 02140
GO TO 360 FCL 02150
349 LD=IFIX(R1) FCL 02160
GO TO 360 FCL 02170
355 WRITE(100UT, 357) FCL 02180
357 FORMAT(1H0,20X,46H***FCLLOUD ERROR*** INPUT RECORD DETECTED WHICH, FCL 02190
+ 44HDOES NOT CORRESPOND TO CORRECT INPUT FORMAT,/) FCL 02200
GO TO 360 FCL 02210
358 WRITE(100UT, 359) FCL 02220
359 FORMAT(1H0,35X,45H***FCLLOUD ERROR*** TOO MANY INPUT CARDS OR GO, FCL 02230
+ 16H SENTINEL ABSENT,/) FCL 02240
IERR=1 FCL 02250
GO TO 200 FCL 02260
360 CONTINUE FCL 02270
361 CONTINUE FCL 02280
ISW=IFLG+IFLO FCL 02290
IF (ISW.EQ.3) ISW=2 FCL 02300
IF (IGEOSW.NE.1) GO TO 222 FCL 02310
XC=PTS(13) FCL 02320
YC=PTS(14) FCL 02330
ZC=PTS(15) FCL 02340
XR=PTS(4) FCL 02350
YR=PTS(5) FCL 02360
ZR=PTS(6) FCL 02370
XS=PTS(1) FCL 02380
YS=PTS(2) FCL 02390
ZS=PTS(3) FCL 02400
222 CONTINUE FCL 02410
C ECHO INPUT FCL 02420
C WRITE(100UT, 1000) XC, YC, ZC, AE, BE, CE, XR, YR, ZR, XS, YS, ZS FCL 02430
C IF (ETA.LT.1 E-20) CALL PFNN(LAMBDA, 0, INDEXP, PFN, ETA) FCL 02440
C WRITE(100UT, 1100) INDEXP, ETA, LAMBDA, KAPPA, W0, TAUBAR, THETA0, PHI0, FCL 02450
C 1RH0, LB0, TMPA, TMPC, LD FCL 02460
C IF (ISW.EQ.2) GO TO 15 FCL 02470
C ISW .NE. 2 INDICATES PRELIMINARY GEOMETRICAL CALCULATIONS TO FCL 02480
C BE PERFORMED: COMPUTE INTERSECTIONS (XM,YM,ZM) AND (XN,YN,ZN) OF FCL 02490
C LINE OF SIGHT WITH CLOUD, SBAR = LENGTH OF PATH FROM SOURCE TO FCL 02500
C RECEIVER, S = LENGTH OF PATH IN CLOUD, AND TRANS = TRANSMITTANCE FCL 02510
C THROUGH CLOUD FCL 02520
C
L=XS-XR FCL 02530
M=YS-YR FCL 02540
N=ZS-ZR FCL 02550
SBAR=SQRT(L*L+M*M+N*N) FCL 02560
L=L/SBAR FCL 02570
M=M/SBAR FCL 02580
N=N/SBAR FCL 02590
DX=XS-XC FCL 02600
DY=YS-YC FCL 02610
DZ=ZS-ZC FCL 02620
ASQ=AE*AE FCL 02630
BSQ=BE*BE FCL 02640
CSQ=CE*CE FCL 02650
ABSQ=ASQ*BSQ FCL 02660
ACSQ=ASQ*CSQ FCL 02670

```

```

BCSQ=BSQ*CSQ          FCL 02710
ABCSQ=ASQ*BSQ*CSQ    FCL 02720
A=BCSQ*L*L+ACSQ*M*M+ABSQ*N*N FCL 02730
B=2.0*(L*DX*BCSQ+M*DY*ACSQ+N*DZ*ABSQ) FCL 02740
C=BCSQ*DX*DX+ACSQ*DY*DY+ABSQ*DZ*DZ-ABCSQ FCL 02750
DISCRM=B*B-4.0*A*C FCL 02760
IF(DISCRM.GE.0.0) GO TO 10 FCL 02770
WRITE(IOUT,1200) FCL 02780
S=0.0 FCL 02790
TRANS=1.0 FCL 02800
LP=0.0 FCL 02810
GO TO 90 FCL 02820
10 SRTDSC=SORT(DISCRM)
TWOA=2.0*A FCL 02830
VPLUS=(-B+SRTDSC)/TWOA FCL 02840
VMINUS=(-B-SRTDSC)/TWOA FCL 02850
XM=XS+VPLUS*L FCL 02860
YM=YS+VPLUS*M FCL 02870
ZM=ZS+VPLUS*N FCL 02880
XN=XS+VMINUS*L FCL 02890
YN=YS+VMINUS*M FCL 02900
ZN=ZS+VMINUS*N FCL 02910
S=SORT((XM-XN)*(XM-XN)+(YM-YN)*(YM-YN)+(ZM-ZN)*(ZM-ZN)) FCL 02920
IF(S.LT.1.E-4) S=0.0 FCL 02930
15 TRANS=EXP(-KAPPA*S) FCL 02940
IF(TMPA.LT.-99.0) GO TO 20 FCL 02950
C TMPA >= -99.0 SPECIFIES THERMAL CALCULATION FCL 02960
C RBAR=(AE+BE+CE)/3.0 FCL 02970
CALL THRMCL(RBAR,W0,TMPA,TMPC,LAMBDA,KAPPA,TRANS,LP) FCL 02980
GO TO 90 FCL 02990
C ISW .NE. 1 SPECIFIES NEW PARAMETERS IN COMMON /OPT/; FCL 03000
ASSOCIATED PRELIMINARY COMPUTATIONS YIELD EXTRATERRESTRIAL FCL 03010
IRRADIANCE E0, COORDINATES (L,M,N) OF UNIT VECTOR FCL 03020
POINTING TO SUN, AND TATM = TRANSMITTANCE OF ATMOSPHERE FCL 03030
ABOVE CLOUD FCL 03040
C 20 IF(ISW.EQ.1) GO TO 30 FCL 03100
CALL ILLUM(LAMBDA,LD,E0) FCL 03110
TH0=THETA0/57.2958 FCL 03120
PH0=PHI0/57.2958 FCL 03130
L0=SINC(TH0)*COSC(PH0) FCL 03140
M0=SINC(TH0)*SINC(PH0) FCL 03150
N0=COSC(TH0) FCL 03160
TATM=EXP(-TAUBAR/N0) FCL 03170
30 WRITE(IOUT,1600) L,M,N,L0,M0,N0 FCL 03180
IF(INDEXP.GT.-1) GO TO 40 FCL 03190
C INDEXP < 0 SPECIFIES MULTIPLE SCATTERING COMPUTATION; FCL 03200
COMPUTE TAU AND TAU0 = CLOUD OPTICAL DEPTH AND THICKNESS, FCL 03210
AND CALL MSCLD FOR VALUE OF LP = PATH RADIANCE FCL 03220
C XG=(XM+XN)/2.0 FCL 03230
YG=(YM+YN)/2.0 FCL 03240
ZG=(ZM+ZN)/2.0 FCL 03250
H=CE*SQRT(1.0-(XG-XC)*(XG-XC)/(AE*AE)-(YG-YC)*(YG-YC)/(BE*BE)) FCL 03260
TAU0=2.0*H*KAPPA FCL 03270
H=ZC-ZG+H FCL 03280
TAU=KAPPA*H FCL 03290
CALL MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP) FCL 03300
GO TO 90 FCL 03310
C INDEXP > -1 SPECIFIES SINGLE SCATTERING COMPUTATION; FCL 03320
COMPUTE SCATTERING ANGLE CHI, PHASE FUNCTION VALUE P = FCL 03330
PFN(LAMBDA,CHI), GEOMETRICAL PARAMETERS ALPHA, BETA, GAMMA, FCL 03340
DELTA, AND EPS, RGRND = GROUND RADIANCE, AND CALL SSCLD FCL 03350
TO COMPUTE PATH RADIANCE FCL 03360
FCL 03370
FCL 03380
FCL 03390
FCL 03400

```

```

40 THETA= ACOS(N)
PHI=0.0
IF(M.NE.0.0.OR.L.NE.0.0) PHI=ATAN2(M,L)
COSCHI=N*N0+SINK(THETA)*SINK(THT0)*COS(PHI-PH0)
CHI= ACOS(COSCHI)*57.2958
CALL PFNN(LAMBDA,CHI,INDEXP,PFN,ETA)
P=PFN
RGRND=E0*TATM*N0
IF(LB0.LT.1.E-10) LB0=RHO*RGRND
DX=XM-XC
DY=YM-YC
DZ=ZM-ZC
LOS=L0*L0
MOS=M0*M0
NOS=N0*N0
LUM=L0*M-M0*L
LUN=L0*N-N0*L
NOM=N0*M-M0*N
MXLY=M0*DX-L0*DY
NXLZ=N0*DX-L0*DZ
MZNY=M0*DZ-N0*DY
DENOM=BCSQ*L0S+ACSQ*MOS+ABSQ*NOS
ALPHA=(BCSQ*L0+ACSQ*M*M0+ABSQ*N*N0)/DENOM
BETA=- $(BCSQ*L0*DX+ACSQ*M0*DY+ABSQ*N0*DZ)/DENOM$ 
GAMMA=- $(BCSQ*(CSQ*L0M*L0M+BSQ*L0N*L0N+ASQ*N0M*N0M))/DENOM$ 
DELTA=CSQ*MXLY*Y*L0M+BSQ*NXLZ*L0N+ASQ*MZNY*NOM
DELTA=-2.0*BCSQ*DELTA/(DENOM*DENOM)
EPS=BCSQ*L0S+ACSQ*MOS+ABSQ*NOS-CSQ*MXLY*MXLY-BSQ*NXLZ*NXLZ
1-ASQ*MZNY*MZNY
EPS=ABCSQ*EPS/(DENOM*DENOM)
IF(EPS.GE.0.0) GO TO 50
WRITE(I00UT,1300) EPS
EPS=0.0
50 IF(GAMMA.LT.0.0) GO TO 60
WRITE(I00UT,1400) GAMMA
GO TO 200
60 CALL SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,W0,RGRND,P,LP)
CCC WRITE RESULTS
90 F=LP/(LB0*TRANS)
TC=1.0/(1.0+F)
WRITE(I00UT,1500) S,TRANS,LP,TC
200 RETURN
1000 FORMAT(1H0,43X,45H-- RADIATIVE TRANSFER THROUGH FINITE CLOUD --/
1 1H0/45X,14H(XC,YC,ZC) = <,2(F8.4,1H),F8.4,12H> KILOMETERS/
2 45X,14H(AE,BE,CE) = <,2(F8.4,1H),F8.4,1H>/
3 45X,14H(XR,YR,ZR) = <,2(F8.4,1H),F8.4,1H>/
4 45X,14H(XS,YS,ZS) = <,2(F8.4,1H),F8.4,1H>
1100 FORMAT(1H0,45X,8HINDEXP = ,I9,10X,8HETA = ,F11.3/
1 45X,8HLAMBDA = ,F9.3,8H <MU> ,2X,9HKAPPA = ,1PE10.4,7H <KM-1>/
2 45X,8HOMEGA 0 = ,0PF9.3,10X,8HTAUBAR = ,1X,1PE10.4/
3 45X,8HTHETA0 = ,0PF9.1,10X,8HPHIO = ,F11.1,10H <DEGREES>/
4 45X,8HRHO = ,0PF9.3,10X,8HLB0 = ,F11.3,13H <W/M2-SR-MU>/
5 45X,8HTMPA = ,F9.1,8H <DEG.C>,2X,8HTMPC = ,F11.1/
6 45X,8HLD = ,I9)
1200 FORMAT(1H0,44X,43H**LINE-OF-SIGHT MISSES CLOUD. S SET TO 0.0>
1300 FORMAT(1H0,44X,4HEPS=,E10.4,24H LT 0.0, EPS SET TO 0.0>
1400 FORMAT(1H0,44X,6HGAMMA=,E10.4,29H GE 0.0, SKIP TO NEXT CASE.>
1500 FORMAT(1H0/1H0,37X,11HPATH LENGTH,3X,15HTRANSMITTANCE ,
1 13HPATH RADIANCE,4X,8HCONTRAST/
2 38X,11H IN CLOUD>,19X,12H<W/M2-SR-MU>,15H TRANSMITTANCE/
3 36X,4<15H+-----+>/
4 1H0,40X,F6.3,7X,1PE9.3,5X,1PE9.3,5X,1PE9.3)
1600 FORMAT(1H0,45X,28HUNIT SOURCE VECTOR L,M,N = ,3<1X,F7.4>/
1 1H0,45X,23HSOLAR VECTOR L0,M0,N0 = ,3<1X,F7.4>)
END

```

```

SUBROUTINE THRMCL(RBAR,W0,TMPA,TMPC,LAMBDA,KAPPA,TRANS,LP)      THR00010
REAL LAMBDA,LP,LI,KAPPA                                         THR00020
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU THR00030
BBC(X,T)=1.19106E8/(X**5*(EXP(1.4388E4/(X*T))-1.0))          THR00040
C*****                                                 THR00050
C SUBROUTINE PERFORMS THERMAL RADIATION CALCULATIONS FOR FINITE THR00060
C CLOUD, RETURNING THE VALUE OF LP = PATH RADIANCE.             THR00070
C INPUTS ARE:                                                 THR00080
C
C   RBAR = AVERAGE HALF-LENGTH OF PATH THROUGH CLOUD           THR00100
C   W0 = SINGLE SCATTERING ALBEDO                            THR00110
C   TMPA = TEMPERATURE OF THE ATMOSPHERE                      THR00120
C   TMPC = TEMPERATURE OF THE CLOUD                          THR00130
C   LAMBDA = WAVELENGTH                                     THR00140
C   KAPPA = VOLUME EXTINCTION COEFFICIENT                  THR00150
C   TRANS = TRANSMITTANCE THROUGH CLOUD.                     THR00160
C
C*****                                                 THR00170
C   G=1.0-EXP(-KAPPA*RBAR)                                THR00180
C   B=BBC(LAMBDA,273.16+TMPC)                           THR00190
C   LI=BBC(LAMBDA,273.16+TMPA)                           THR00200
C   WRITE(I0OUT,1000) B,LI                               THR00210
C   LP=(1.0-TRANS)*(1.0-W0)*(1.0+G*W0)*B+W0*(1.0-G)*LI>    THR00220
1000 FORMAT(1H0,43) ,40H **THERMAL CALCULATION OF PATH RADIANCE/,45X, THR00230
122H   BBC(LAMBDA,TMPC) = ,1PE10.4,11H W/M2-SR-MU/,45X,        THR00240
122H   BBC(LAMBDA,TMPA) = ,1PE10.4,11H W/M2-SR-MU>          THR00250
      RETURN                                              THR00260
      END                                                 THR00270
                                                THR00280
                                                THR00290

```

```

SUBROUTINE SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,           SSC00010
1W0,RGRND,P,LP)                                                 SSC00020
COMMON /IOUNIT/IOUT,IIN,IPHFUN,LOUNT,NDIRTU,NCLIMT,KSTOR,NPLOTUSSC00030
*****SSC00040
*****SSC00050
*****SSC00060
*****SSC00070
*****SSC00080
*****SSC00090
*****SSC00100
*****SSC00110
*****SSC00120
*****SSC00130
*****SSC00140
*****SSC00150
*****SSC00160
*****SSC00170
*****SSC00180
*****SSC00190
*****SSC00200
*****SSC00210
*****SSC00220
*****SSC00230
*****SSC00240
*****SSC00250
*****SSC00260
*****SSC00270
*****SSC00280
*****SSC00290
*****SSC00300
*****SSC00310
*****SSC00320
*****SSC00330
*****SSC00340
*****SSC00350
*****SSC00360
*****SSC00370

```

SUBROUTINE COMPUTES PATH RADIANCE LP FOR THE CASE OF SINGLE SCATTERING IN A FINITE CLOUD. INPUTS ARE GEOMETRICAL PARAMETERS ALPHA, BETA, GAMMA, DELTA, AND EPS, S = LENGTH OF OPTICAL PATH THROUGH CLOUD, AND:

KAPPA = VOLUME EXTINCTION COEFFICIENT
W0 = SINGLE SCATTERING ALBEDO
RGRND = SURFACE BACKGROUND RADIANCE
P = SINGLE SCATTERING PHASE FUNCTION.

REAL KAPPA,LP
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDECK
ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))
WRITE(IOUT,1500)
DISCRM=DELTA*DELTA-4.0*GAMMA*EPS
IF(DISCRM.GT.0.0) GO TO 10
WRITE(IOUT,1000) DISCRM
LP=0.0
GO TO 20
10 TGSDEL=2.0*GAMMA*S+DELTA
HS= ASIN(TGSDEL/SQRT(DISCRM))/SQRT(-GAMMA)
H0= ASIN(DELTA/SQRT(DISCRM))/SQRT(-GAMMA)
TRANS=EXP(-KAPPA*S)
BETAB=W0*KAPPA
LP=(1.0-BETA*KAPPA)*S+(1.0-ALPHA)*KAPPA*S*S/2.0
LP=LP-KAPPA*DISCRM*(HS-H0)/(8.0*GAMMA)
LP=LP-KAPPA*(TGSDEL+SQRT(GAMMA*S*S+DELTA*S+EPS))-
1DELTA*SQRT(EPS))/(4.0*GAMMA)
LP=BETAB*LP*P*RGRND*TRANS/(4.0*PI)
20 RETURN
1000 FORMAT(9H0DISCRM =,E10.4,25H IN SSCLD, LP SET TO 0.0)
1500 FORMAT(1H0,43X,33H **RESULTS FOR SINGLE SCATTERING)
END

```

SUBROUTINE PFNN(LAMBDA,CHI,INDEXP,PFN,ETA) PFN00010
REAL LAMBDA,LAMDA1,LAM1,LAM2,KAPPA PFN00020
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNT,NDIRTU,NCLIMT,KSTOR,NPLOTUP FN00030
COMMON /CONST/PI,PI2,PIRAD,TWOP,TORRMB,CDEGK PFN00040
DIMENSION PHANG(65),PF1(65),PF2(65) PFN00050
DATA LAMDA1,IP1,CHI1/-1.0,-2,-1.0/ PFN00060
C***** PFN00070
C SUBROUTINE COMPUTES THE VALUE OF THE INDEXP-TH PHASE FUNCTION PFN00080
C AT SCATTERING ANGLE CHI AND WAVELENGTH LAMBDA USING BILINEAR PFN00090
C INTERPOLATION. PFN00100
C PFN00110
C PFN00120
C PFN00130
ACOS(ARG)=ATAN2(SQRT(1,-ARG**2),ARG)
IERR=0 PFN00140
MAXID=12
C PFNDAT ONLY CONTAINS ONE PFN FOR THE VISIBLE (.55UM) AND PFN00150
C THE NEAR IR (< 1.06UM); THEREFORE DO NOT INTERPOLATE.
IWAVE=0 PFN00160
IF (LAMBDA.LT.2.0) IWAVE=1 PFN00170
PRELIMINARIES FOR FINDING ETA PFN00180
ETA1=0.0 PFN00190
ETA2=0.0 PFN00200
IDCK=0 PFN00210
IF (INDEXP.LT.0) IDCK=1 PFN00220
IF (INDEXP.LT.0) INDEXP=-INDEXP PFN00230
IF (INDEXP.EQ.IP1.AND.CHI.EQ.CHI1.AND.LAMBDA.EQ.LAMDA1) PFN00240
100 TO 200 PFN00250
IF (INDEXP.EQ.IP1.AND.LAMBDA.EQ.LAMDA1) GO TO 70 PFN00260
REWIND IPHFUN PFN00270
NRD=0 PFN00280
DO 20 I=1,66,11 PFN00290
READ(IPHFUN,1000) (PHANG(I+J-1),J=1,11) PFN00300
DO 10 J=1,11 PFN00310
NRD=NRD+1 PFN00320
IF (PHANG(NRD).GE.999.9) GO TO 30 PFN00330
10 CONTINUE PFN00340
20 CONTINUE PFN00350
30 NA=NRD-1 PFN00360
DO 35 I=1,NA PFN00370
PHANG(I)=COS(PHANG(I)*PIRAD) PFN00380
CONTINUE PFN00390
C END-OF-FILE CHECK PFN00400
IF (IERR.EQ.2) GO TO 195 PFN00410
READ(IPHFUN,1100) IANG1, ID, LAM1, W0, KAPPA, BETA PFN00420
IF (LAM1.GE.12.00.AND.ID.EQ.MAXID) IERR=2 PFN00430
READ(IPHFUN,1200) (PF1(I),I=1,NA) PFN00440
SUM=0.
C START RENORMALIZATION OF PHASE FUNCTION - ALSO SEE BELOW PFN00450
DO 45 J=2,NA PFN00460
SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF1(J)+PF1(J-1))/4. PFN00470
DO 46 J=1,NA PFN00480
PF1(J)=PF1(J)/SUM PFN00490
ETA1=ETAIN(T(PF1,PHANG,NA)) PFN00500
IF (ID.NE.INDEXP) GO TO 40 PFN00510
IF (IWAVE.EQ.1.AND.LAMBDA.GT.LAM1) GO TO 40 PFN00520
IF (IWAVE.EQ.1) GO TO 75 PFN00530
IF (LAMBDA.LT.LAM1) GO TO 190
50 CONTINUE PFN00540
IF (IERR.EQ.2) GO TO 195
READ(IPHFUN,1100) IANG2, ID, LAM2, W0, KAPPA, BETA
IF (LAM2.GE.12.00.AND.ID.EQ.MAXID) IERR=2
READ(IPHFUN,1200) (PF2(I),I=1,NA)
SUM=0.
DO 55 J=2,NA
SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF2(J)+PF2(J-1))/4.
DO 56 J=1,NA
PF2(J)=PF2(J)/SUM
ETA2=ETAIN(T(PF2,PHANG,NA))
C THE PHASE FUNCTION(S) ARE NOW NORMALIZED TO: INTEGRAL OF

```

```

C      PHASE FUNCTION OVER ALL SOLID ANGLE DIVIDED BY 4 PI = 1.          PFN00550
IF(ID,NE,INDEXP) GO TO 190                                              PFN00560
IF(LAMBDA,LE,LAM2) GO TO 70                                              PFN00570
LAM1=LAM2
IANG1=IANG2
DO 60 I=1,NA
PF1(I)=PF2(I)
60 CONTINUE
GO TO 50
CONTINUE
IF (IWAVE.EQ.1) GO TO 75                                              PFN00640
DLAM=(LAMBDA-LAM1)/(LAM2-LAM1)
75 CONTINUE
IF(CHI.LT.-1.E-3.OR.CHI.GT.180.001) GO TO 190
DO 80 J=2,NA
IF(CHI.LE.(ACOS(PHANG(J))/PIRAD)) GO TO 90
80 CONTINUE
J=NA
90 J1=J-1
DCHI=(COS(CHI*PIRAD)-PHANG(J1))/(PHANG(J)-PHANG(J1))
IF (IWAVE,NE,1) GO TO 95
PFN=PF1(J1)+DCHI*(PF1(J)-PF1(J1))
95 PFN=PF1(J1)+DLAM*(PF2(J1)-PF1(J1))+DCHI*(PF1(J)-PF1(J1))
1+DLAM*DCHI*(PF2(J)+PF1(J1)-PF2(J1)-PF1(J))
96 LAMDA1=LAMBDA
CHI1=CHI
IF (IP1,NE,-2) WRITE(100UT,1500) CHI,PFN
IP1=INDEXP
ETA=ETA1+DLAM*(ETA2-ETA1)
GO TO 200
190 WRITE(100UT,1300) ID,INDEXP,LAM1,LAMBDA,CHI
STOP
195 WRITE (100UT,1600) IPHFUN
STOP
200 CONTINUE
IF (IDCK.EQ.1) INDEXP=-INDEXP
RETURN
1000 FORMAT(1I<F6.2,1X>)
1100 FORMAT(2<I2,1X>,F5.2,1X,F8.6,1X,2(E12.6,1X))
1200 FORMAT(6(E12.6,1X))
1300 FORMAT(33H0ERROR IN READING PHASE FUNCTION./
127H ID,INDEXP,LAM1,LAMBDA,CHI=,2I3,3E13.7)
1500 FORMAT(1H0,23HSCATTERING ANGLE CHI = ,F8.2,7X,24H PHASE FN P(LAMBDA)PFN00900
+A,CHI)=,E10.4)
1600 FORMAT(1X,32HATTEMPT TO READ PAST EOF ON UNIT,I3,18H IN SUBROUTINEPFN00920
1_PFN//)
END

```

```

SUBROUTINE MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP)      MSC00010
REAL LP,LBAR,K                                              MSC00020
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK               MSC00030
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUMSC00040
SINH(ARG)=.5*(EXP(ARG)-EXP(-ARG))
COSH(ARG)=.5*(EXP(ARG)+EXP(-ARG))
*****MSC00050
C SUBROUTINE COMPUTES LP = PATH RADIANCE DUE TO MULTIPLE SCATTERING MSC00060
C IN A FINITE CLOUD. INPUTS ARE:                                MSC00080
MSC00090
TAU = CLOUD OPTICAL DEPTH                                     MSC00100
TAU0 = CLOUD OPTICAL THICKNESS                                MSC00110
TRANS = TRANSMITTANCE ALONG LINE OF SIGHT                   MSC00120
TATM = TRANSMITTANCE OF ATMOSPHERE ABOVE CLOUD              MSC00130
E0 = EXTRATERRESTRIAL IRRADIANCE                            MSC00140
W0 = SINGLE SCATTERING ALBEDO                             MSC00150
ETA = FORWARD SCATTERING PARAMETER                         MSC00160
RHO = BACKGROUND SURFACE REFLECTANCE.                      MSC00170
MSC00180
*****MSC00190
IF(W0.LT.1.0) GO TO 10
W0=.999
WRITE(1000)
10 C1=1.0-ETA*W0
C2=(1.0-ETA)*W0
K=SQRT((1.0-W0)*(1.0+W0-2.0*ETA*W0))
TOK=K*TAU0
TOTK=K*(TAU0-TAU)
TK2=2.0*K*TAU
GAMMA=E0*TATM/(C1*SINH(TOK)+K*COSH(TOK))
DELTA=RHO*K/((C1-RHO*C2)*SINH(2.0*TOK)+K*COSH(2.0*TOK))
WRITE(1000)
EPLUS=C2*SINH(TOK)+DELTA*(C1*SINH(TK2)+K*COSH(TK2))
EMINUS=C1*SINH(TOK)+K*COSH(TOK)+DELTA*C2*SINH(TK2)
LBAR=GAMMA*(EPLUS+EMINUS)/TWOP1
LP=W0*LBAR*(1.0-TRANS)
WRITE(1000)TAU,TAU0,TATM
100 FORMAT(1H0,45X,18H OPTICAL DEPTH = ,1PE10.4/,45X,
131H OPTICAL THICKNESS OF CLOUD = ,1PE10.4/,45X,
244H TRANSMITTANCE OF ATMOSPHERE ABOVE CLOUD = ,1PE10.4)
RETURN
1000 FORMAT(36H0 **RESULTS FOR MULTIPLE SCATTERING )
2000 FORMAT(47H0 **OMEGA 0 WAS 1.0, NOW SET TO 0.999 IN MSCLD)
END

```

```
C FUNCTION ETAINTR(PFN,PHANG,NA)
C THIS FUNCTION WILL DETERMINE ETA, THE FORWARD SCATTERING
C PARAMETER: ETA=.5*INTEGRAL PFN OVER THETA, WHERE THETA GOES
C FROM ZERO TO PI/2.
C COMMON /CONST/ PI,PI2,PIRAD,TWOPi,TORRMB,CDEGK
C DIMENSION PHANG(65),PFN(65)
C NAM1=NA-1
C ETA=0.
C DO 1 I=1,NAM1
C IF(PHANG(I+1).GT.0.) ETA=ETA+(PHANG(I)-PHANG(I+1))*1
C (PFN(I+1)+PFN(I))/4.
C 1 CONTINUE
C ETAINTR=ETA
C RETURN
C END
```

```

SUBROUTINE OVRCS( LAMBDA, TRANS, IERR )
REAL LC,LG,LB0,KAPPA,LAMBDA,MU,LB
DIMENSION IALPH(7)
COMMON /IOUNIT/IIN, IOUT, IPHFUN, LUNIT, NDINTU, NCLINT, KSTOR, NPLOTUDVR00010
COMMON /GEOMET/PTS(15),IGEOSW VR00020
DATA IZERO/0/ VR00030
DATA IALPH/2H0 ,2HSP ,2HCL ,2HBK ,2HGR ,2HTE ,2HGD / VR00040
BB(X,T)=1.191062E8/(X**5*(EXP(1.4387864E4/(X*T))-1.0)) VR00050

```

SUBROUTINE COMPUTES BEAM TRANSMITTANCE, PATH RADIANCE, AND
CONTRAST TRANSMITTANCE ALONG AN OPTICAL PATH UNDER AN OVERCAST
SKY. ORDER-INDEPENDENT INPUT CARDS ARE AS FOLLOWS:
(INDIVIDUAL RECORD FORMAT IS (A4,1X,5(E10.4,1X)))

MNEMONIC	VARIABLES READ	DESCRIPTION	OVR
OPOS	XO,YO,ZO	OBSERVER POSITION	000160
SPOS	XT,YT,ZT	SOURCE POSITION	000170
CCLS	ZC,LC,KAPPA,ETA,W0	CLOUD PARAMETERS	000180
BKGR	LB0	BACKGROUND RADIANCE	000190
GRND	LG	GROUND RADIANCE	000200
TEMP	TEMP	TEMPERATURE ALONG PATH	000210
GO		END OF READ SENTINEL	000220

THE VARIABLES ZC AND LC REFER TO THE OVERCAST SKY LAYER!
THE VARIABLES KAPPA, ETA, W0, TEMP, REFER TO THE ATMOSPHERE
BETWEEN THE OVERCAST SKY AND GROUND I.E. THE INTERVENING
ATMOSPHERIC PROPERTIES (GAS OR AEROSOL)

** NOTE : THE GO CARD MUST BE THE LAST RECORD READ.

THE FOLLOWING ENUMERATES THE VARIABLES LISTED ON THE ABOVE CARDS :

(XO,YO,ZO) = OBSERVER COORDINATES	OVR00370
(XT,YT,ZT) = SOURCE COORDINATES	OVR00380
ZC = HEIGHT OF CLOUD LAYER	OVR00390
LC = CLOUD RADIANCE	OVR00390
LG = GROUND RADIANCE	OVR00390
LB0 = BACKGROUND RADIANCE	OVR00390
KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1)	OVR00390
ETA = FORWARD SCATTERING PARAMETER	OVR00390
W0 = SINGLE SCATTERING ALBEDO	OVR00390
TEMP = TEMPERATURE ALONG PATH (DEG. C)	OVR00400

LENGTH UNITS ARE KILOMETERS; RADIANCE UNITS ARE W/M2-SR-MU,
IF TEMP >= -99., THERMAL RADIATION IS CALCULATED; IF TEMP < -99.
SINGLY SCATTERED RADIATION IS CALCULATED.

SUBROUTINE RETURNS:
TRANS = BEAM TRANSMITTANCE
TO CALLING PROGRAM.

DATA INITIALIZATION

```

IF(IZERO.NE.0) GO TO 477
XO=0.0 OVR00510
YO=0.0 OVR00520
ZO=0.0 OVR00530
XT=0.0 OVR00540
YT=0.0 OVR00550
ZT=0.0 OVR00560
ZC=0.0 OVR00570
LC=0.0 OVR00580
KAPPA=0.0 OVR00590
ETA =0.0 OVR00600
W0=0.0 OVR00610
LB0=0.0 OVR00620
OVR00630
OVR00640
OVR00650
OVR00660

```

```

LG=0.0
TEMP=0.0
IZERO=1
477 CONTINUE
DO 360 K=1,7
READ(10IN,334)IA,IA2,R1,R2,R3,R4,R5
334 FORMAT(2A2,1X,5(E10.4,1X))
DO 333 I=1,8
IF(IA.NE.IALPH(I)) GO TO 333
IND=I
IF(IND.EQ.7) GO TO 361
333 CONTINUE
IF(IND.EQ.8) GO TO 355
IF(K.EQ.7.AND.IND.NE.7) GO TO 358
GO TO (341,342,343,344,345,346),IND
341 X0=R1
Y0=R2
Z0=R3
GO TO 360
342 XT=R1
YT=R2
ZT=R3
GO TO 360
343 ZC=R1
LC=R2
KAPPA=R3
ETA=R4
W0=R5
GO TO 360
344 LB0=R1
GO TO 360
345 LG=R1
GO TO 360
346 TEMP=R1
GO TO 360
355 WRITE(10OUT,357)
357 FORMAT(1H0,25X,44H***OVRCSST ERROR*** INPUT CARD DETECTED WHICH,
+ 36H DOES NOT MATCH CORRECT INPUT FORMAT,/),
GO TO 360
358 WRITE(10OUT,359)
359 FORMAT(1H0,34X,45H***OVRCSST ERROR*** TOO MANY INPUT CARDS OR GO,
+ 16H SENTINEL ABSENT,/),
IERR=1
GO TO 200
360 CONTINUE
361 CONTINUE
IF(IGEOSW.NE.1) GO TO 222
X0=PTS(4)
Y0=PTS(5)
Z0=PTS(6)
XT=PTS(1)
YT=PTS(2)
ZT=PTS(3)
222 CONTINUE
C ECHO INPUT
C
1 WRITE(10OUT,1000) X0,XT,Y0,ZT,Z0,ZC,LC,LAMBDA,LG,
1 TEMP,LB0,KAPPA,W0,ETA
IF(TEMP.LT.-99.0) GO TO 4
BBTEMP=BB(LAMBDA,273.16+TEMP)
WRITE(10OUT,1800) BBTEMP
BTE=(W0-1.0)*BBTEMP
GO TO 8
4 WRITE(10OUT,1700)
8 F=2.0*(1.0-ETA)
ZLEN=ZT-Z0
S=SQRT((XT-X0)**2+(YT-Y0)**2+ZLEN**2)
MU=ABS(ZLEN)/S
TO=KAPPA*Z0

```

```

TT=KAPPA*ZT
SO=KAPPA*(ZC-ZD)
ST=KAPPA*(ZL-ZT)
TRANS=EXP(-KAPPA*S)
CONST=W0*ETA /2.0
IF(ZLEN) 10,40,70

C HEIGHT OF OBSERVER > HEIGHT OF SOURCE
10 G21=G2(T0,TT,MU)
G22=G2(S0,ST,-MU)
T1=EXP(-T0/MU)
T2=EXP(S0/MU)
F1=W0*F*KAPPA*S*T1/2.0
XINT1=CONST*T1*G21
XINT2=CONST*T2*G22
IF(TEMP.GE.-99.0) GO TO 20
LP=LG*(F1+XINT1)+LC*XINT2
GO TO 100
20 LP=(LG+BTE)*(F1+XINT1)+(LC+BTE)*XINT2+W0*BTE*(TRANS-1.0)
GO TO 100

C HEIGHT OF OBSERVER = HEIGHT OF SOURCE
40 E21=1.0
E22=1.0
IF(T0.NE.0.0) E21=EXP(-T0)-T0*E1(T0)
IF(S0.NE.0.0) E22=EXP(-S0)-S0*E1(S0)
IF(TEMP.GE.-99.0) GO TO 50
LP=CONST*(1.0-TRANS)*(LG*E21+LC*E22)
GO TO 100
50 LP=CONST*(E21*(LG+BTE)+E22*(LC+BTE))
LP=(LP-W0*BTE)*(1.0-TRANS)
GO TO 100

C HEIGHT OF OBSERVER < HEIGHT OF SOURCE
70 G21=G2(T0,TT,-MU)
G22=G2(S0,ST,MU)
T1=EXP(-S0/MU)
T2=EXP(T0/MU)
F1=W0*F*KAPPA*S*T1/2.0
XINT1=CONST*T2*G21
XINT2=CONST*T1*G22
IF(TEMP.GE.-99.0) GO TO 80
LP=LC*(F1+XINT2)+LG*XINT1
GO TO 100
80 LP=(LC+BTE)*(F1+XINT2)+(LG+BTE)*XINT1+W0*BTE*(TRANS-1.0)

C WRITE RESULTS
100 TC=1.0/(1.0+LP/(LB0*TRANS))
WRITE(I00UT,2000) S,TRANS,LP,TC
200 RETURN
1000 FORMAT(1H0/1H0,55X,34H-- RADIATION UNDER OVERCAST SKY --/
1 1H0,43X,8HXO      =,5X,F6.3,5H <KM>,8X,5HXT  =,5X,F6.3,5H <KM>/
2 44X,8HYO      =,5X,F6.3,13X,5HYT  =,5X,F6.3/
3 44X,8HZO      =,5X,F6.3,13X,5HZT  =,5X,F6.3/
4 44X,8HLC      =,5X,F6.3,13X,5HLC  =,1X,1PE10.4,13H <W/M2-SR-MU>/
5 44X,8HLAMBDA =,5X,0PF6.3,5H <MU>,8X,5HLG  =,1X,1PE10.4/
6 44X,8HTEMP    =,3X,0PF6.1,10H <6EG.C>,5X,5HLB0 =,1X,1PE10.4/
7 44X,8HKAPPA   =,1X,E10.4,7H <KM-1>,6X,5HW0  =,5X,0PF5.3/
8 44X,8HETA     =,6X,0PF5.3)
2000 FORMAT(1H0,39X,4HPATH,7X,13HTRANSMITTANCE,2X,13HPATH RADIANCE,
1 4X,8HCONTRAST/
2 38X,11HLENGTH <KM>,19X,12H<W/M2-SR-MU>,15H TRANSMITTANCE/
3 36X,4(15H+-----+)
4 1H0,39X,F6.3,9X,F7.5,7X,1PE9.3,7X,0PF7.5)
1700 FORMAT(1H0,50X,31H**RESULTS FOR SINGLE SCATTERING)
1800 FORMAT(1H0,46X,38H**THERMAL CALCULATION OF PATH RADIANCE/

```

1 1H0,50X,8HBBTEMP= ,1PE10.4,11H W/M2-SR-MU>
END

OVR02070
OVR02080

```

FUNCTION G2(TAU1,TAU2,MU) FUG00010
C***** FUG00020
C SUBROUTINE COMPUTES IN CLOSED FORM INTEGRALS OF THE FUNCTION FUG00030
C EXP(TAU/MU)*E2(TAU) FUG00040
C WHERE E2 IS THE SECOND EXPONENTIAL INTEGRAL. FOR DETAILS SEE FUG00050
C KOURGANOFF, 'BASIC METHODS IN TRANSFER PROBLEMS', APPENDIX I FUG00060
C (PAGES 256-257 OF FIRST EDITION, 1952, OXFORD UNIVERSITY PRESS.) FUG00070
C***** FUG00080
C REAL MU FUG00090
C DATA GAMMA/.5772156649/ FUG00100
C IF(MU.LT..9999) GO TO 50 FUG00110
C C MU EQ 1.0 FUG00120
C IF(TAU1.NE.0.0) GO TO 10 FUG00130
C G2=(TAU2-1.0)*EXP(TAU2)*E1(TAU2)-GAMMA- ALOG(TAU2) FUG00140
C GO TO 100 FUG00150
C 10 IF(TAU2.NE.0.0) GO TO 20 FUG00160
C G2=GAMMA+ALOG(TAU1)+EXP(TAU1)*(1.0-TAU1)*E1(TAU1) FUG00170
C GO TO 100 FUG00180
C 20 G2=EXP(TAU1)*(1.0-TAU1)*E1(TAU1)-EXP(TAU2)*(1.0-TAU2)*E1(TAU2) FUG00190
C G2=G2+(ALOG(TAU1)-ALOG(TAU2)) FUG00200
C GO TO 100 FUG00210
C C MU NE 1.0 FUG00220
C 50 RM=1.0-1.0/MU FUG00230
C IF(TAU1.NE.0.0) GO TO 60 FUG00240
C G2=EXP(TAU2/MU)*(TAU2-MU)*E1(TAU2) FUG00250
C G2=G2+MU*E1(TAU2*RM) FUG00260
C G2=G2+1.0+MU*ALOG(ABS(RM))-EXP(-RM*TAU2) FUG00270
C GO TO 100 FUG00280
C 60 IF(TAU2.NE.0.0) GO TO 70 FUG00290
C G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1) FUG00300
C G2=G2-MU*E1(TAU1*RM) FUG00310
C G2=G2+EXP(-RM*TAU1)-MU*ALOG(ABS(RM))-1.0 FUG00320
C GO TO 100 FUG00330
C 70 G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1) FUG00340
C G2=G2-EXP(TAU2/MU)*(MU-TAU2)*E1(TAU2) FUG00350
C G2=G2+(EXP(-TAU1*RM)-EXP(-TAU2*RM)) FUG00360
C G2=G2+MU*(E1(TAU2*RM)-E1(TAU1*RM)) FUG00370
C 100 RETURN FUG00380
C END FUG00390

```

```

FUNCTION E1(X)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU FUE00010
FUE00020
***** FUE00030
FUNCTION COMPUTES THE VALUE OF THE FIRST EXPONENTIAL INTEGRAL FUE00040
-EI(-X) WHERE -174 < X < 170. FOR X OUTSIDE THESE BOUNDS, AN FUE00050
OVERFLOW OR UNDERFLOW MIGHT OCCUR, SO PROGRAM EXECUTION IS FUE00060
HALTED. POLYNOMIAL AND RATIONAL FUNCTION APPROXIMATIONS ARE FUE00070
ADAPTED FROM THE IBM SCIENTIFIC SUBROUTINE PACKAGE, SUBROUTINE FUE00080
EXPI. FUE00090
FUE00100
FUE00110
***** FUE00120
DATA GAMMA/.5772156649/ FUE00130
DATA C1,C2,C3,C4,C5,C6,C7,C8/674.567029,57.411833,6.05529232, FUE00140
11699.06552,841.654932,49.3133893,8.01957683,.99979204/ FUE00150
DATA D1,D2,D3,D4,D5,D6,D7,D8,D9/248.6697,224.4234,32.43665, FUE00160
13.061037,.05176245,180.7837,22.63818,38.93944,3.995161/ FUE00170
DATA F1,F2,F3,F4,F5,F6,F7,F8,F9/9.999999E-1,2.500001E-1, FUE00180
15.555682E-2,1.041576E-2,1.664156E-3,2.335379E-4,2.928433E-5, FUE00190
21.766345E-6,7.122452E-7/ FUE00200
DATA G1,G2,G3,G4,G5,G6,G7,G8/.2677737343,8.6347608925, FUE00210
118.059016973,8.5733287401,3.9584969228,21.0996530827, FUE00220
225.6329561486,9.5733223454/ FUE00230
IF(X.GT.-174.0.AND.X.LE.170.0) GO TO 10 FUE00240
WRITE(100UT,1000) FUE00250
STOP FUE00260
10 IF(X.GT.-9.0) GO TO 20 FUE00270
E1=1.0-(<C1+C2*X-C3*X*X-X*X*X)/(<C4+C5*X+C6*X*X-C7*X*X*X-C8*X*X*X*X) FUE00280
E1=E1*EXP(-X)/X FUE00290
GO TO 100 FUE00300
20 IF(X.GT.-3.0) GO TO 30 FUE00310
E1=D1+D2*X+D3*X*X+D4*X*X*X+D5*X*X*X*X FUE00320
E1=<(1.0-E1/(<D6+D7*X+D8*X*X+D9*X*X*X+X*X*X*X))*EXP(-X)/X FUE00330
GO TO 100 FUE00340
30 IF(X.GT.1.0) GO TO 40 FUE00350
E1=F1-X*<(F2-X*<(F3-X*<(F4-X*<(F5-X*<(F6-X*<(F7-X*<(F8-X*X*F9)))))))>>>>>> FUE00360
E1=X*X*E1-GAMMA- ALOG(ABS(X)) FUE00370
GO TO 100 FUE00380
40 E1=<(G1+X*<(G2+X*<(G3+X*<(G4+X))))/<(G5+X*<(G6+X*<(G7+X*<(G8+X))))>>>>>> FUE00390
E1=E1*EXP(-X)/X FUE00400
100 RETURN FUE00410
1000 FORMAT(4H0X =,3X,E10.4,39H OUT-OF-RANGE FOR E1. EXECUTION HALTED.) FUE00420
END FUE00430

```

```

SUBROUTINE GRNADE(WAVE1,ICLMAT,TRANS,IERR)          GRN00010
C/*****                                         *****/GRN00020
C/*          SUBROUTINE GRNADE                  *GRN00030
C/*          MAIN GRNADE MODULE                *GRN00040
C/*          EOSAEL80                           */GRN00050
C/*****                                         *****/GRN00060
C/*****                                         *****/GRN00070
C/*****                                         *****/GRN00080
C/*****                                         *****/GRN00090
C/*****                                         *****/GRN00100
C/*****                                         *****/GRN00110
C/*****                                         *****/GRN00120
C/*****                                         *****/GRN00130
C/*****                                         *****/GRN00140
C/*****                                         *****/GRN00150
C/*****                                         *****/GRN00160
C/*****                                         *****/GRN00170
C/*****                                         *****/GRN00180
C/*****                                         *****/GRN00190
C*** THE FOLLOWING VARIABLES SUPPLIED BY THE USER.   GRN00200
C*** FIELD DATA                                     GRN00210
C*** XNORTH    FIELD COORDINATES FROM NORTH (DEGREES) GRN00220
C*** HEAD      GRENADE TANK HEADING CLOCKWISE FROM NORTH (DEGREES) GRN00230
C*** RNC       DISTANCE OF GRENADES FROM TANK (METERS) GRN00240
C*** DLEN      GRENADE SPACING <PERPENDICULAR TO HEADING> GRN00250
C*** XO,YO,ZO  COORDINATES OF OBSERVER (METERS) GRN00260
C*** XM,YM,ZM  COORDINATES OF TANK (METERS) GRN00270
C*** XT,YT,ZT  TARGET COORDINATES (METERS) GRN00280
C*** METEORLOGICAL DATA                         GRN00290
C*** WINDP    WIND PROFILE EXPONENT               GRN00300
C*** HM       HEIGHT OF INVERSION LAYER           GRN00310
C*** WD       WIND DIRECTION FROM NORTH (DEGREES) GRN00320
C*** WS       WIND SPEED (METERS PER SECOND)      GRN00330
C*** RH       RELATIVE HUMIDITY                   GRN00340
C*** ICAT     PASQUILL CATEGORY                 GRN00350
C*** YF       SMOKE YIELD FACTOR                 GRN00360
C*** MUNITION DATA                            GRN00370
C*** EFF      CLOUD-MAKING EFFICIENCY OF MUNITION GRN00380
C*** QMUN    TOTAL MASS OF SMOKE AGENT (GRAMS)  GRN00390
C*** DETECTOR DATA                          GRN00400
C*** WAVE1    WAVELENGTH OF INTEREST (MICRONS)  GRN00410
C*** DIFFUSION PARAMETERS                   GRN00420
C*** SIG2     REFERENCE SIGMA (METERS)           GRN00430
C*** XREF    REFERENCE DISTANCE (METERS)         GRN00440
C*** ZDIFF    VERTICAL DIFFUSION CONSTANT        GRN00450
C*** YDIFF    CROSSWIND DIFFUSION CONSTANT       GRN00460
C*** HK      TERRAIN SCAVENGING COEFFICIENT     GRN00470
C*** VS      PARTICLE SETTLING VELOCITY (CM/SEC) GRN00480
C*** RC      TERRAIN REFLECTION COEFFICIENT     GRN00490
C***** DEFINITIONS OF OTHER VARIABLES***          GRN00500
C*** IXMAX   NUMBER OF POINTS ALONG LINE-OF-SIGHT FOR CL COMPUTATION GRN00510
C*** NBPT    NUMBER OF GRENADE LINES (NBPT=1)      GRN00520
C*** NTARG   NUMBER OF TARGETS (NTARG=1)           GRN00530
C*** TBURST  MUNITION DETONATION TIME            GRN00540
C***          MUNITION DETONATION TIME            GRN00550
C***          MUNITION DETONATION TIME            GRN00560
C*** ERROR CODES AND OPTION CODES:              GRN00570
C*** IWRIT=1  DEPRESSES RAW DATA PRINTOUT        GRN00580
C*** IFLAG=4  INVALID DATA CARD (BUT IGNORED)    GRN00590
C*** IFLAG=3  OVER 11 DATA CARDS ENTERED BEFORE GO GRN00600
C***          REMAINDER IGNORED                  GRN00610
C*** IFLAG=2  NORMAL READ TERMINATION           GRN00620
C*** IFLAG=1  WAVELENGTH OF INTEREST NOT IN DEFINED BANDS GRN00630
C***          TRANS SET TO 1.0                  GRN00640
C*****SET DEFAULTS*****                         GRN00650
C***          TBURST=0.0                         GRN00660
C***          CALL GOGET(WAVE1,KWAVE)             GRN00670
C*****READ DATA AND WRITE HEADING*****          GRN00680
C***          WRITE(IOUT,8000)                   GRN00690
C***          8000 FORMAT(1H0,20X,40(2H**),/,21X,1H*,34X,14HPROGRAM GRNADE,30X,1H*,/,GRN00700

```

```

*21X,1H*,37X,BHEOSAEL80,33X,1H*,/,21X,40<2H**)
IWRIT=1 GRN00710
IFLAG=0 GRN00720
6 CALL DATRD(IWRIT,IFLAG) GRN00730
IF(IFLAG.EQ.4)GO TO 9999 GRN00740
C*****CALCULATE INTEGRATION INCREMENT GRN00750
XX=(XT-X0)**2+(YT-Y0)**2+(ZT-Z0)**2 GRN00760
DLOS=SORT(XX) GRN00770
IXMAX=IFIX(DLOS) GRN00780
IF(IXMAX.GT.1000)IXMAX=1000 GRN00790
C*****DEFAULT TO CLIMATE DATA OPTION GRN00800
IF(ICLMAT.NE.1)GO TO 12 GRN00810
RHA=RH GRN00820
UW=WNDVEL GRN00830
WD=WNDDIR GRN00840
ICAT=IFASCT GRN00850
12 CONTINUE GRN00860
IF(DLEN.EQ.0.0)DLEN=10.0 GRN00870
IF(BRATE.EQ.0.0)BRATE=(1.0/14.3) GRN00880
QLENTH=XN*DLEN GRN00890
BREXP=BRATE GRN00900
QMUN=XN*FW GRN00910
EMUN=QMUN*YF*EFF/100.0 GRN00920
WS=UW GRN00930
IF(XMIS<1).LE.0.0)GO TO 4 GRN00940
XREFZ=100.0 GRN00950
SIG2R=XMIS(1) GRN00960
ZD1FF=XMIS(2) GRN00970
YD1FF=XMIS(3) GRN00980
HM=XMIS(4) GRN01000
HK=XMIS(5) GRN01010
RC=XMIS(6) GRN01020
VS=XMIS(7) GRN01030
GO TO 5 GRN01040
4 CALL PARMs(ICAT) GRN01050
5 CONTINUE GRN01060
C*****REDEFINE WIND PROFILE EXPONENT IF READ IN POSITIVE GRN01070
IF(WPOWR.GE.0.0)WINDP=WPOWR GRN01080
IF(WS.LE.0.0) WS=0.1 GRN01090
U2=WS GRN01100
C----- LOCATE GENERATING LINE GRN01110
C----- TRANSFORM TO OBSERVER COORDINATES GRN01120
C----- TRANSFORM TO ORIGIN UNDER OBSERVER AND X-AXIS UNDER TARGET. GRN01130
C----- GET THE WIND DIRECTION ANGLE WITH THE NEW X-AXIS. GRN01140
C----- WRITE INPUT DATA AND HEADINGS GRN01150
C----- WRITE(I001,8001) GRN01160
C----- WRITE(I001,1000) GRN01170
C----- WRITE(I001,1001)WS,WD,ICAT,RHA,XNORTH GRN01180

```

```

      WRITE( IODOUT, 998)
      WRITE( IODOUT, 995)
      WRITE( IODOUT, 997) X0, EXTC(1), Y0, EXTC(2), Z0, EXTC(3),
      *XT, EXTC(4), YT, EXTC(5), ZT, EXTC(6), EXTC(7)
      WRITE( IODOUT, 1002)
      WRITE( IODOUT, 1003) XM, SIGZR, YM, XREFZ, ZM, HM
      WRITE( IODOUT, 1004) HEAD, HK, RNG, RC, XN, VS
      WRITE( IODOUT, 1005) QMUH, WINDP, QLENTH, ZDIFF, BREXP, YDIFF
      WRITE( IODOUT, 1006) EFF, YF
8001 FORMAT( //, 21X, 36H*****INPUT*****ALL LENGTHS IN METERS.,,
*21X, 38(2H--))
1000 FORMAT( 21X, 15HMETEOROLOGICAL:)
1001 FORMAT( 24X, 10HWIND SPEED, 10X, F6.1, 1X, 3HM/S,
      *          24X, 14HWIND DIRECTION, 6X, F6.1, 1X, 3HDEG, /,
      *          24X, 17HPASQUILL CATEGORY, 3X, I3, /,
      *          24X, 17HRELATIVE HUMIDITY, 3X, F6.1, 1X, 1HZ, /,
      *          24X, 21HNOTE: X AXIS HEADING, 1X, F6.1, 1X, 3HDEG, 1X,
      *          28HCLOCKWISE FROM NORTH (DCWFN))
1002 FORMAT( 20X, 19HTANK/MUNITION DATA:, 16X, 21HDIFFUSION PARAMETERS:)
1003 FORMAT( 24X, 7HXX TANK), 12X, F6.1, 10X, 10HSIGZ(XREF), 21X, F6.1, /,
      *          24X, 7HY(TANK), 12X, F6.1, 10X, 4HXREF, 27X, F6.1, /,
      *          24X, 7HZ(TANK), 12X, F6.1, 10X, 17HMIXING HEIGHT(HM), 14X, F6.1,
1004 FORMAT( 24X, 14HHEADING(DCWFN), 5X, F6.1, 10X, 20HSCAVENGING COEFF(HK),
      *11X, F6.3, /, 24X, 5HRANGE, 14X, F6.1, 10X, 20HRÉFLECTION COEFF(RC), 11X,
      *F6.3, /, 24X, 10HNO GRNADES, 9X, F6.1, 10X, 21HSETTLING VELOCITY(VS),
      *5X, F6.3, 1X, 4HCM/S)
1005 FORMAT( 24X, 14HSMOKE MASS(GM), 5X, F6.1, 10X, 29HVERTICAL WIND EXPONENT
      *(WPOWR), 2X, F6.3, /, 24X, 11HLINE LENGTH, 8X, F6.1, 10X,
      *29HVERTICAL DIFF CONSTANT(ZDIFF), 2X, F6.3, /,
      *24X, 13HBURN CONSTANT, 2X, F6.3, 1X, 3H1/S, 10X,
      *30HCROSSWIND DIFF CONSTANT(YDIFF), 1X, F6.3)
1006 FORMAT( 24X, 10HEFFICIENCY, 9X, F6.1, 10X, 12HYIELD FACTOR, 17X, F6.1)
998 FORMAT( 21X, 16HOBSERVER/TARGET:, 8X, 24HEXTINCTION COEFFICIENTS:)
995 FORMAT( 49X, 7HMICRONS, 3X, 7HM**2/GM)
997 FORMAT( 24X, 6HX(OBS), 3X, F6.1, 10X, 7H0.4-0.7, 3X, F6.3, /,
      *          24X, 6HY(OBS), 3X, F6.1, 10X, 7H0.7-1.2, 3X, F6.3, /,
      *          24X, 6HZ(OBS), 3X, F6.1, 10X, 7H1.06, 3X, F6.3, /,
      *          24X, 6HX(TAR), 3X, F6.1, 10X, 7H3.0-5.0, 3X, F6.3, /,
      *          24X, 6HY(TAR), 3X, F6.1, 10X, 7H8.0-12., 3X, F6.3, /,
      *          24X, 6HZ(TAR), 3X, F6.1, 10X, 7H10.6, 3X, F6.3, /,
      *          49X, 4H94.0, 1X, 3GHZ, 2X, F6.3)
      WRITE( IODOUT, 996)
996 FORMAT( 1H1, 21X, 16H*****OUTPUT*****/, 21X, 38(2H--))
      WRITE( IODOUT, 3000)
3000 FORMAT( 24X, 4HTIME, 6X, 2HCL, 23X, 12HTRANSMISSION, /, 24X, 5H(SEC), 2X,
      *9H(GM/M**2), 2X, 7H0.4-0.7, 1X, 7H0.7-1.2, 3X, 4H1.06, 2X, 7H3.0-5.0, 1X,
      *7H8.0-12., 2X, 4H10.6, 4X, 5H94GHZ)
C----- BEGIN CL CALCULATIONS -----
C----- DO 400 IT=IST0, IETO, IDTO
        DO 400 IT=IST0, IETO, IDTO
        ITT=IT
C     * SET UP LOOP ON SPACIAL DISTRIBUTION
        XC = 0.0
        YC = 0.0
        ZC = 20
        DELX=SQRT( (XT-X0)**2+(YT-Y0)**2 )/IXMAX
        DELZ=(ZT-Z0)/IXMAX
C** FOR EACH TIME GET THE CONCENTRATION AT IXMAX POINTS ALONG LINE-OF-SIGHT
        DO 300 IX=1, IXMAX
          XC = XC + DELX
          ZC = ZC + DELZ
          XA(1) = XC
          XA(2) = YC
          XA(3) = ZC
          T = FLOAT(IT)
          UBXB=U2
          CALL CONCN(XA,T,C)
          XDTAC(IX)=XC
          CDTAC(IX)=C
        GRN01410
        GRN01420
        GRN01430
        GRN01440
        GRN01450
        GRN01460
        GRN01470
        GRN01480
        GRN01490
        GRN01500
        GRN01510
        GRN01520
        GRN01530
        GRN01540
        GRN01550
        GRN01560
        GRN01570
        GRN01580
        GRN01590
        GRN01600
        GRN01610
        GRN01620
        GRN01630
        GRN01640
        GRN01650
        GRN01660
        GRN01670
        GRN01680
        GRN01690
        GRN01700
        GRN01710
        GRN01720
        GRN01730
        GRN01740
        GRN01750
        GRN01760
        GRN01770
        GRN01780
        GRN01790
        GRN01800
        GRN01810
        GRN01820
        GRN01830
        GRN01840
        GRN01850
        GRN01860
        GRN01870
        GRN01880
        GRN01890
        GRN01900
        GRN01910
        GRN01920
        GRN01930
        GRN01940
        GRN01950
        GRN01960
        GRN01970
        GRN01980
        GRN01990
        GRN02000
        GRN02010
        GRN02020
        GRN02030
        GRN02040
        GRN02050
        GRN02060
        GRN02070
        GRN02080
        GRN02090
        GRN02100

```

```

300 CONTINUE          GRN02110
C*** INTEGRATE ALONG LINE-OF-SIGHT TO GET CL.   GRN02120
    CALL SUMA (XDTA,CDTA, CL ,IXMAX)   GRN02130
C----- GRN02140
C----- GRN02150
C----- GRN02160
C----- GRN02170
DO 500 J=1,7          GRN02180
  XTRAN(J)=EXP(-1.0*EXTC(J)*CL(IXMAX))   GRN02190
500 CONTINUE          GRN02200
  WRITE( IOOUT,2000>T,CL(IXMAX),(XTRAN(J),J=1,7)   GRN02210
2000 FORMAT(23X,F5.1,2X,F8.3,2X,7(1X,F7.3))   GRN02220
  400 CONTINUE          GRN02230
C***** GRN02240
C*****SET TRANSMISSION FOR RETURN TO EOSAEL   GRN02250
C***** IF(KWAVE.GT.0)GO TO 3   GRN02260
  TRANS=1.0   GRN02270
  IFLAG=1   GRN02280
  3 TRANS=XTRAN(KWAVE)   GRN02290
  GO TO 6   GRN02300
9999 WRITE( IOOUT,9000)   GRN02310
9000 FORMAT(21X,28H****PROGRAM GRNADE END****)   GRN02320
  RETURN   GRN02330
END               GRN02340

```

```

SUBROUTINE CONCN(XA,T,C2)                                     COC00010
C/***** SUBROUTINE CONCN                                         COC00020
C/*          GRNAD MODULE                                         COC00030
C/*          EOSAEL80                                           COC00040
C/***** PURPOSE:                                                 COC00050
C/*          CALCULATES CONCENTRATION AT A SPECIFIED POSITION AND TIME.   COC00060
C/*          THE CONCENTRATION IS USED WITH THE EXTINCTION COEFFICIENT    COC00070
C/*          TO COMPUTE TRANSMITTANCE.                                         COC00080
C/*          DESCRIPTION OF PARAMETERS:                                       COC00090
C/*          XA      - POSITION IN METERS, INPUT.                         COC00100
C/*          T       - TIME IN SECONDS, INPUT.                           COC00110
C/*          C2      - CONCENTRATION, OUTPUT.                          COC00120
C/*          SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:                 COC00130
C/*          LOCAT      UMEAN                                         COC00140
C/*          COMMON BLOCK STATEMENTS REQUIRED:                            COC00150
C/*          MECH1     MECH2     MECH3                                         COC00160
C/*          REMARKS:                                              COC00170
C/*          CONCN COMPUTES FIVE TERMS AND MULTIPLIES THEM TO           COC00180
C/*          GET CONCENTRATION.                                         COC00190
C/***** DIMENSION XA(3)                                         COC00200
COMMON/CONST/PI,PI2,PIRAD,TWOPi,TORRMb,CDECK                COC00250
COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP            COC00260
COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH                         COC00270
COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM        COC00280
C----- CONCENTRATION=TERM1*TERM2*TERM3*TERM4*TERMS             COC00290
C----- C2 = 0,0                                                 COC00300
C----- X = XA(1)                                              COC00310
C----- Y = XA(2)                                              COC00320
C----- Z = XA(3)                                              COC00330
C*** GET CROSSWIND AND DOWNWIND COMPONENTS OF GENERATING LINE. COC00340
QC=ABS(QLENTH*COS(THETA))                                     COC00350
QD=ABS(QLENTH*SIN(THETA))                                     COC00360
NBPT=1                                                       COC00370
DO 390 J = 1,NBPT                                         COC00380
TJ = T- TTI(J)                                              COC00390
C----- TERM1 IS CLOUD MASS AS A FUNCTION OF TIME.              COC00400
C----- IF(BREXP*TJ,GE,200,0)STOP01                            COC00410
TRM1=EMUN*(1.0-EXP(-BREXP*TJ))                                COC00420
C----- TERM2 IS TERRAIN SCAVENGING TERM.                      COC00430
C----- TERM3 IS DOWNWIND PROBABILITY DENSITY.                  COC00440
C*** CHANGE TO SMOKE COORDINATES.                            COC00450
CALL LOCAT(J,X,Y,Z,XB,YB,ZB)                                 COC00460
C*** UPWIND END OF GENERATING LINE WILL BE ORIGIN OF SMOKE SYSTEM. COC00470
XB=XB+0.5*QD                                              COC00480
CALL UMEAN(J,TJ)                                            COC00490
IF(HK*XB/UBXB,GE,200,0)STOP01                                COC00500
TRM2 = EXP(-HK*XB/UBXB)                                      COC00510
C----- TERM4 IS CROSSWIND PROBABILITY DENSITY.                  COC00520
C----- UT=UBXB*TJ                                             COC00530
IF(XB,LE,0.) GO TO 999                                         COC00540
IF(XB,GT,0.,AND.XB,LE,UT) TERM3=1./                COC00550
IF(XB,GT,UT,AND.XB,LE,UT+QD) TERM3=(UT+QD-XB)/(UT*QD+0.5*QD*QD) COC00560
IF(XB,GT,UT+QD) TERM3=0.                                         COC00570
TRM3=TERM3                                              COC00580
C----- TERM4 IS CROSSWIND PROBABILITY DENSITY.                  COC00590
C----- YWIDTH=YDIFF*XB+QC                                     COC00600
TRM4= 1./YWIDTH                                         COC00610

```

```

IF(YB .GT. 0.5*YWIDTH) TRM4=0,
IF( YB .LT. -0.5*YWIDTH) TRM4=0,
C----- TERMS IS VERTICAL PROBABILITY DENSITY.
C----- SIGZT=SIGZR*(XB/XREFZ)**2DIFF
C----- S2T2=SIGZT*SIGZT
C----- HMVT = Z(J) - (VS/100.0)*(XB/UBXB)
C----- T1 = EXP(- $(HMVT-ZB)^2/(2.0*S2T2)$ )
C----- T2 = RC*EXP(- $(HMVT+ZB)^2/(2.0*S2T2)$ )
C----- TERR = 1.0E-06
C----- J0 = 0
C----- T1 = 0.0
C----- T2 = 0.0
C----- T3 = 0.0
C----- T4 = 0.0
C----- TS = 0.0
370 JC = JC+1
R1 = RC** $(JC-1)$ 
R2 = RC**JC
R3 = RC** $(JC+1)$ 
IF( <2.0*JC*HM+HMVT+ZB>**2, GT, 600, *2,*S2T2) GO TO 371
T1 = R1*EXP(- $(2.0*JC*HM-HMVT-ZB)^2/(2.0*S2T2)$ )
T2 = R2*EXP(- $(2.0*JC*HM-HMVT+ZB)^2/(2.0*S2T2)$ )
T3 = R2*EXP(- $(2.0*JC*HM+HMVT-ZB)^2/(2.0*S2T2)$ )
T4 = R3*EXP(- $(2.0*JC*HM+HMVT+ZB)^2/(2.0*S2T2)$ )
T1234 = T1 + T2 + T3 + T4
TS = TS + T1234
IF(T1234 - TERR) 371,371,370
371 CONTINUE
TV3 = TS
TRM5 = TV1+TV2+TV3
TRM5=TRM5/ $SQRT(2.*PI)$ 
C1 = TRM1*TRM2*TRM3*TRM4*TRM5
C2 = C2 + C1
380 CONTINUE
999 RETURN
END

```

```

SUBROUTINE GOGET(WAVE1,KWAVE)                               GOG00010
C/*****SUBROUTINE GOGET                                         GOG00020
C/*          SUBROUTINE GOGET                                     */GOG00030
C/*          GRNAD MODULE                                       */GOG00040
C/*          EOSAEL80                                         */GOG00050
C/*****SUBROUTINE FINDS SPECTRAL BAND FOR GIVEN SINGLE WAVELENGTH GOG00060
C*****SUBROUTINE FINDS SPECTRAL BAND FOR GIVEN SINGLE WAVELENGTH GOG00070
      KWAVE=0                                                 GOG00080
      IF(WAVE1.GE.0.40.AND.WAVE1.LT.0.70)KWAVE=1             GOG00090
      IF(WAVE1.GE.0.70.AND.WAVE1.LT.1.20)KWAVE=2             GOG00100
      IF(WAVE1.GE.1.20.AND.WAVE1.LT.1.50)KWAVE=3             GOG00110
      IF(WAVE1.GE.1.50.AND.WAVE1.LT.5.00)KWAVE=4             GOG00120
      IF(WAVE1.GE.5.00.AND.WAVE1.LT.12.0)KWAVE=5             GOG00130
      IF(WAVE1.EQ.1.06)                                     KWAVE=3             GOG00140
      IF(WAVE1.EQ.10.6)                                     KWAVE=6             GOG00150
      IF(WAVE1.EQ.94.0)                                      KWAVE=7             GOG00160
      IF(WAVE1.GT.3188.0.AND.WAVE1.LT.3195.0)KWAVE=7         GOG00170
      RETURN
      END

```

```

SUBROUTINE PARMs(ICAT)                                PAR00010
C/*****SUBROUTINE PARMs                               PAR00020
C/*          GRNAD MODULE                           PAR00030
C/*          EOSAEL80                                PAR00040
C/*****SETS DIFFUSION PARAMETER DEFAULTS AS FUNCTION OF PASQUILL CATEGORYPAR00050
C/*****COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM      PAR00060
COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM      PAR00070
YDIFF=0.355                                              PAR00080
HK=0.002                                                PAR00090
VS=0.021                                                PAR00100
RC=0.70                                                 PAR00110
XREFZ=100.0                                              PAR00120
IF(ICAT.GT.3)GO TO 1                                  PAR00130
WINDP=0.10                                              PAR00140
ZDIFF=2.08                                              PAR00150
HM=1000.0                                               PAR00160
SIGZR=14.0                                              PAR00170
GO TO 3                                                 PAR00180
1 IF(ICAT.GT.4)GO TO 2                                PAR00190
WINDP=0.20                                              PAR00200
ZDIFF=1.40                                              PAR00210
HM=300.0                                                PAR00220
SIGZR=7.2                                              PAR00230
GO TO 3                                                 PAR00240
2 WINDP=0.40                                              PAR00250
ZDIFF=1.04                                              PAR00260
HM=50.0                                                 PAR00270
SIGZR=5.0                                              PAR00280
3 RETURN                                              PAR00290
END                                                    PAR00310

```

```
SUBROUTINE EXTN(EX)
C*****SUBROUTINE EXTN
C/*          GRNAD MODULE
C/*          EOSAEL80
C*****PROGRAM TO SET EXTINCTION COEFFICIENTS FOR WP/RP SMOKE*****
C*****DIMENSION EX(7),CX(7)
      SUBROUTINE EXTN
      DATA CX/4.304,2.166,1.541,0.350,0.338,0.364,0.001/
      DO 1 I=1,7
      EX(I)=CX(I)
1   CONTINUE
      RETURN
      END
```

EXTN0010
EXTN0020
EXTN0030
EXTN0040
EXTN0050
EXTN0060
EXTN0070
EXTN0080
EXTN0090
EXTN0100
EXTN0110
EXTN0120
EXTN0130
EXTN0140

```

SUBROUTINE UMEAN(J,TJ)                                UME00010
C/*****SUBROUTINE UMEAN                               UME00020
C/*          GRNAD MODULE                           **/UME00030
C/*          EOSAEL80                                **/UME00040
C/*
C/*****CALCULATES MEAN WIND SPEED OVER EXTENT OF CLOUD UME00050
COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP      UME00060
COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH                UME00070
COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM UME00080
IC = 0                                                 UME00090
QD = ABS(QLENTH*SIN(THETA))                          UME00100
P = WINDP + 1.0                                       UME00110
C1=P*2.0**WINDP                                      UME00120
Z2=ZI(J)                                              UME00130
IF(Z2<2.0)10,10,20                                     UME00140
10 CONTINUE                                           UME00150
Z2=2.0                                               UME00160
Z1=0.01                                             UME00170
GO TO 200                                            UME00180
20 ZI=ZI(J)-1.5*SIGZR                                UME00190
IF(Z1>30,30,40)                                     UME00200
30 Z1=0.01                                           UME00210
40 CONTINUE                                           UME00220
UH = (U2/((Z2-Z1)*C1))*((Z2**P-Z1**P)             UME00230
US=UH                                              UME00240
50 CONTINUE                                           UME00250
XC = UH*TJ                                         UME00260
Z2 = ZI(J) - (VS/100.0)*(XC/UH)                      UME00270
IF(Z2<2.0)151,151,155                                UME00280
151 CONTINUE                                           UME00290
Z2=2.0                                               UME00300
Z1=0.01                                             UME00310
GO TO 160                                            UME00320
155 CONTINUE                                           UME00330
SIGZT = SIGZR*((XC+QD)/XREFZ)**ZDIFF               UME00340
Z1 = ZI(J) - 1.5*SIGZT                                UME00350
IF(Z1>159,159,160)                                     UME00360
159 CONTINUE                                           UME00370
Z1=0.01                                             UME00380
160 UBXJ = (U2/((Z2-Z1)*C1))*((Z2**P-Z1**P)         UME00390
GO TO 1000                                           UME00400
200 CONTINUE                                           UME00410
US=U2                                              UME00420
UH=US                                              UME00430
UBXJ=US                                             UME00440
1000 CONTINUE                                           UME00450
UBXJ = SQRT((UBXJ**2 + UH**2)/2.0)                  UME00460
IC = IC + 1                                         UME00470
IF(IC.EQ.1) UH = UBXJ                                UME00480
IF(IC.EQ.1) GO TO 50                                UME00490
UBXB=UBXJ                                           UME00500
RETURN                                              UME00510
END                                                 UME00520
                                         UME00530
                                         UME00540

```

```

SUBROUTINE SUMA(X,Y,Z,NDIM)                                SUMA0010
C/*****                                                       SUMA0020
C/*          SUBROUTINE SUMA                               */ SUMA0030
C/*          GRNAD MODULE                               */ SUMA0040
C/*          EOSAEL80                                 */ SUMA0050
C/*****                                                       SUMA0060
C/*                                                       SUMA0070
C/* PURPOSE:                                              SUMA0080
C/* GENERAL PURPOSE INTEGRATION SUBROUTINE               */ SUMA0090
C/* USAGE:                                                 SUMA0100
C/* CALLED FROM GRNAD TO GET INTEGRAL OF CONCENTRATION ALONG */ SUMA0110
C/* LINE-OF-SIGHT.                                         */ SUMA0120
C/* DESCRIPTION OF PARAMETERS:                           */ SUMA0130
C/* X           INDEPENDENT VARIABLE                   */ SUMA0140
C/* Y           DEPENDENT VARIABLE                    */ SUMA0150
C/* Z           INTEGRAL OF Y OVER X                  */ SUMA0160
C/* NDIM        NUMBER OF POINTS                     */ SUMA0170
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:      */ SUMA0180
C/* NONE                                                 */ SUMA0190
C/* COMMON BLOCK STATEMENTS REQUIRED:                 */ SUMA0200
C/* NONE                                                 */ SUMA0210
C/* REMARKS:                                             */ SUMA0220
C/* AS USED BY GRENADE, X IS POSITION ONLINE-OF-SIGHT, Y IS */ SUMA0230
C/* LOCAL CONCENTRATION, Z IS TOTAL CONCENTRATION.       */ SUMA0240
C/* METHOD:                                              */ SUMA0250
C/* SUMA INCREMENTS INTEGRAL BY AVERAGE OF LAST TWO Y VALUES */ SUMA0260
C/* TIMES DELTA X.                                         */ SUMA0270
C/*****                                                       SUMA0280
DIMENSION X(1000),Y(1000),Z(1000)                         SUMA0290
SUM2 = 0.0                                                    SUMA0300
IF (NDIM - 1) 4,3,1                                       SUMA0310
C 1 DO 2 I = 2,NDIM                                     SUMA0320
  SUM1 = SUM2                                              SUMA0330
  SUM2 = SUM2 + 0.500*(X(I)-X(I-1))*(Y(I)+Y(I-1))    SUMA0340
23  Z(I-1) = SUM1                                         SUMA0350
  Z(NDIM) = SUM2                                         SUMA0360
4   RETURN                                                 SUMA0370
END                                                       SUMA0380
                                                       SUMA0390

```

```

SUBROUTINE LOCAT(I,X,Y,Z,XB,YB,ZB)                                LOC00010
C*****SUBROUTINE LOCAT*****                                         LOC00020
C/* SUBROUTINE LOCAT                                              */ LOC00030
C/* GRNAD MODULE                                                 */ LOC00040
C/* EOSAEL80                                                    */ LOC00050
C*****PURPOSE:                                                 */ LOC00060
C/* LOCAT TRANSLATES POSITION INTO BURST AND WIND COORDINATES. */ LOC00070
C/* USAGE:                                                       */ LOC00080
C/* CALLED BY CONCN                                           */ LOC00090
C/* DESCRIPTION OF PARAMETERS:                                 */ LOC00100
C/* T               TIME (SECONDS), INPUT                         */ LOC00110
C/* X,Y,Z           ORIGINAL POSITION (METERS), INPUT          */ LOC00120
C/* XI,YI,ZI        ORIGINAL BURST POSITION (METERS), INPUT    */ LOC00130
C/* WDA            ANGLE BETWEEN WIND VECTOR AND LINE OF SIGHT */ LOC00140
C/*                  (RADIAN), INPUT                           */ LOC00150
C/* XB,YB,ZB        TRANSLATED POSITION (METERS), OUTPUT       */ LOC00160
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:             */ LOC00170
C/* NONE                                                       */ LOC00180
C/* COMMON BLOCK STATEMENTS REQUIRED:                         */ LOC00190
C/* MECH1           MECH2                                         */ LOC00200
C/* REMARKS:                                                   */ LOC00210
C/* ORIGINAL COORDINATES HAVE ORIGIN AT OBSERVER AND X-AXIS */ LOC00220
C/* THROUGH TARGET. NEW COORDINATES HAVE ORIGIN AT BURST      */ LOC00230
C/* AND X-AXIS IN DIRECTION OF WIND VECTOR.                   */ LOC00240
C/* METHOD:                                                   */ LOC00250
C/* STANDARD ROTATION AND TRANSLATION OF AXES                */ LOC00260
C*****COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP          LOC00270
CCOMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH                          LOC00280
CLOC00290
XB = (X - XI(I))*COS(WDA) + (Y - YI(I))*SIN(WDA)           LOC00300
LOC00310
YB = -(X - XI(I))*SIN(WDA) + (Y - YI(I))*COS(WDA)           LOC00320
LOC00330
ZB=Z
IF(ZB.LT.0.0.AND.ABS(ZB).GT.ZI(I)) ZB=0.0                  LOC00340
LOC00350
RETURN
END

```

```

SUBROUTINE DATRD(IWRIT,IFLAG)
C*****SUBROUTINE DATRD
C***      GRNAD MODULE
C***      EUSAE80
C*****THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS
C   THE SMOKE(EUSAE) PROGRAM
INPUTS
EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,
FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.
PER FIELD BEGINNING IN COL 11. THE CARDS ARE NOT ORDER
DEPENDENT.

NAME          IGNORED
BURN          IGNORED
MUNC          XM, YM, ZM
              HEAD    COORDINATES OF GRENADE FIRING TANK
                      HEADING OF GRENADE FIRING TANK CLOCKWISE
                      FROM NORTH
              RNG     GRENADE FIRING RANGE OF TANK
              DLEN    SPACING OF GRENADES ALONG LINE PERPENDICULAR
                      TO HEADING
OBSC          X0, Y0, Z0
TARC          XT, YT, ZT
BART          ST0    COORDINATES OF THE OBSERVER (M,M,M)
              ETO    COORDINATES OF THE TARGET (M,M,M)
              DTO
              XNORTH
HUNT           XN    STARTING TIME (ELAPSED TIME SINCE BLAST)
                  ENDING TIME FOR CALCULATION
                  TIME INCREMENT FOR CALCULATION
                  X AXIS HEADING CLOCKWISE FROM NORTH
              FW
              TBURN
              ITYPE
EFF            EFF    NUMBER OF MUNITIONS FIRED AT THE SAME
                  LOCATION AND AT THE SAME TIME
                  FILL WEIGHT (LBS)
                  BURN TIME OF SMOKE TYPE (IGNORED)
                  TYPE OF SMOKE (DEFAULTS TO 1)
                  1.=WP, 2.=PWP, 3.=HC, 4.=FOG OIL
                  EFFICIENCY OF BURN (PERCENT). IF 0.0,
                  DEFAULTS TO 62.0%
YF              YF    YIELD FACTOR IF 0.0, DEFAULTS TO ANALYTICAL
                  MODEL
METR           BRATE
RHA            RHA    EXPONENTIAL BURN RATE PARAMETER
UW             UW
WD             WD
ICAT           ICAT
AIRT           AIRT
TGRAD          TGRA
WPOWR          WPOW
EXTC           WPOW
              DESIRED CHANGES IN EXTINCTION COEFF.
              (OPTIONAL), IF NOT USED OR READ AS 0.
              DEFAULTS TO ALPHA ARRAY VALUE IN STRNS.
              BANDS ARE:
                  0.4-0.7 MICRONS
                  0.7-1.2 MICRONS
                  1.06 MICRONS
                  3.0-5.0 MICRONS
                  8.0-12. MICRONS
                  10.6 MICRONS
                  94.0 GHZ.
MISC           SIGZR
              ZDIFF
              YDIFF
              HM
              HK
              RC
              VS
GO             GO
DONE           DONE
              DIFFUSION PARAMETER OPTION CARD FOR GRNAD
              DOWNWIND REFERENCE AT 100 M REFERENCE DIST.
              VERTICAL DIFFUSION COEFFICIENT
              CROSSWIND DIFFUSION COEFFICIENT
              HEIGHT OF MIXING LAYER (METERS)
              TERRAIN SCAVENGING COEFFICIENT
              TERRAIN REFLECTION COEFFICIENT
              SETTLING VELOCITY (CM SEC)
              SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT
              END OF JOB.

```

```

***** COMMON /IOUNIT/I0IN, I0OUT, IPHFUN, L0UNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUDAT00710
COMMON /GEOMET/PTS(15),IGEOSW DAT00720
COMMON/CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB, DAT00730
      WNDVEL,WNDDIR,IPASCT DAT00740
* COMMON/MECH0/XM,YM,ZM,X0,Y0,Z0,XT,YT,ZT,ISTO,IETO,ITO,XN,FW, DAT00750
*T BURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG, DAT00760
*DLEN,WPOWR,EXTC(8),XMIS(8),XNORTH DAT00770
DIMENSION DUMY(8),IR(26),IR1(10),R1(10),EX(7),INAME(35) DAT00780
DATA IR/2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA, DAT00790
*2HRC,2HEX,2HTC,2HBU,2HBN,2HMI,2HSC,2HGD,2H ,2HDO,2HNE,2HNA,2HME, DAT00800
*2HOU,2HTP/
IF (IFLAG.GT.0) GO TO 8 DAT00810
DO 2 J=1,8 DAT00820
XMIS(J)=0.0 DAT00830
2 EXIT(J)=0. DAT00840
DAT00850
DAT00860
DAT00870
DAT00880
DAT00890
DAT00900
DAT00910
DAT00920
FORMAT(1H0,21X,20H****CARD INPUT****,/,21X,40(2H--)) DAT00930
FORMAT(2A2,6X,35A2) DAT00940
FORMAT(1H0,21X,2A2,6X,35A2) DAT00950
6 DO 70 I=1,13 DAT00960
IF(I.EQ.13) GO TO 310 DAT00970
IF(IFLAG.GT.0) GO TO 4 DAT00980
IFLAG=1 DAT00990
READ(I0IN,201)IR1(1),IR1(2),(INAME(J),J=1,35) DAT01000
IF(IWRIT.EQ.0) GO TO 4 DAT01010
WRITE(I0OUT,202)IR1(1),IR1(2),(INAME(J),J=1,35) DAT01020
4 READ(I0IN,203)IR1(1),IR1(2),(R1(J),J=2,8) DAT01030
IF(IWRIT.EQ.0) GO TO 5 DAT01040
WRITE(I0OUT,30)IR1(1),IR1(2),(R1(J),J=2,8) DAT01050
5 IF(IR1(1).EQ.IR(21).AND.IR1(2).EQ.IR(22)) GO TO 998 DAT01060
20 FORMAT(2A2,6X,7F10.3) DAT01070
30 FORMAT(1H0,21X,2A2,6X,7F10.3) DAT01080
DAT01090
***** RELATING INPUT DATA TO VARIABLE NAMES. DAT01100
***** DAT01110
IF(IR1(1).EQ.IR(1).AND.IR1(2).EQ.IR(2)) GO TO 90 DAT01120
IF(IR1(1).EQ.IR(3).AND.IR1(2).EQ.IR(4)) GO TO 100 DAT01130
IF(IR1(1).EQ.IR(5).AND.IR1(2).EQ.IR(6)) GO TO 110 DAT01140
IF(IR1(1).EQ.IR(7).AND.IR1(2).EQ.IR(8)) GO TO 120 DAT01150
IF(IR1(1).EQ.IR(9).AND.IR1(2).EQ.IR(10)) GO TO 130 DAT01160
IF(IR1(1).EQ.IR(11).AND.IR1(2).EQ.IR(12)) GO TO 140 DAT01170
IF(IR1(1).EQ.IR(13).AND.IR1(2).EQ.IR(14)) GO TO 150 DAT01180
IF(IR1(1).EQ.IR(15).AND.IR1(2).EQ.IR(16)) GO TO 155 DAT01190
IF(IR1(1).EQ.IR(17).AND.IR1(2).EQ.IR(18)) GO TO 165 DAT01200
IF(IR1(1).EQ.IR(19).AND.IR1(2).EQ.IR(20)) GO TO 175 DAT01210
IF(IR1(1).EQ.IR(21).AND.IR1(2).EQ.IR(22)) GO TO 998 DAT01220
IF(IR1(1).EQ.IR(23).AND.IR1(2).EQ.IR(24)) GO TO 70 DAT01230
IF(IR1(1).EQ.IR(25).AND.IR1(2).EQ.IR(26)) GO TO 70 DAT01240
DAT01250
***** ERROR CAUTION FOR INVALID DATA CARD DAT01260
***** DAT01270
IFLAG=2 DAT01280
WRITE(I0OUT,80) DAT01290
80 FORMAT(21X,35H****CAUTION**** INVALID DATA CARD) DAT01300
GO TO 70 DAT01310
90   RHA    = R1(2) DAT01320
     UW     = R1(3) DAT01330
     WD     = R1(4) DAT01340
     ICAT   = IFIX(R1(5)) DAT01350
     AIRT   = R1(6) DAT01360
     TGRAD  = R1(7) DAT01370
     WPOWR=R1(8) DAT01380
100  GO TO 70 DAT01390
     XN     = R1(2) DAT01400

```

```

FW      =453.6*R1(3)          DAT01410
TBURN   =R1(4)               DAT01420
ITYPE   =IFIX(R1(5))         DAT01430
EFF     =R1(6)               DAT01440
YF=R1(7)                      DAT01450
BRATE=R1(8)                    DAT01460
GO TO 70                         DAT01470
110    ISTO   = IFIX(R1(2))    DAT01480
      IETO   = IFIX(R1(3))    DAT01490
      IDTO   = IFIX(R1(4))    DAT01500
      XNORTH= R1(5)           DAT01510
      IF(ISTO.LE.0)ISTO=1       DAT01520
      DAT01530
      GO TO 70                 DAT01540
120    XM     = R1(2)           DAT01550
      YM     = R1(3)           DAT01560
      ZM     = R1(4)           DAT01570
      HEAD   = R1(5)           DAT01580
      RNG    = R1(6)           DAT01590
      DLEN   = R1(7)           DAT01600
      DAT01610
      GO TO 70                 DAT01620
130    XU     = R1(2)           DAT01630
      YO     = R1(3)           DAT01640
      ZU     = R1(4)           DAT01650
      DAT01660
      GO TO 70                 DAT01670
140    XT     = R1(2)           DAT01680
      YT     = R1(3)           DAT01690
      ZT     = R1(4)           DAT01700
      DAT01710
C*****BURN CARD DATA DUMMYED BY PROGRAM GRNAD*****
C*****DO 156 J=1,7             DAT01720
156    DUMY(J)=R1(J+1)         DAT01730
      GO TO 70                 DAT01740
150    DO 152 J=1,7             DAT01750
152    EXTC(J)=R1(J+1)         DAT01760
      GO TO 70                 DAT01770
165    DO 166 J=1,7             DAT01780
166    XMIS(J)=R1(J+1)         DAT01790
      DAT01800
      70 CONTINUE
      175 GO TO 311             DAT01810
C*****CAUTION FOR TOO MANY CARDS
C*****310 WRITE(IQOUT,320)
      IFLAG=3                  DAT01820
      DAT01830
      DAT01840
      DAT01850
C*****DEFAULT NON USER DEFINED INPUT*****
C*****311 IF(ITYPE.EQ.1)GO TO 3
      ITYPE=1                  DAT01860
      WRITE(IQOUT,171)           DAT01870
      DAT01880
      DAT01890
      DAT01900
      DAT01910
      171 FORMAT(1H,21X,17H*****CAUTION****/,1H .21X,54HWRONG SMOKE TYPE
      *FOR PROGRAM GRNAD--DEFAULTED TO WP/RP)  DAT01920
      3 IF(EFF.EQ.0.0)EFF=62.0        DAT01930
      IF(YF.EQ.0.0)YF=3.14+0.032*RHA  DAT01940
      IF(EXTC(1).GT.0.0)GO TO 1       DAT01950
      CALL EXTIN(EX)
      DO 7 I=' '
      7 EXTC(I)=EX(I)                DAT01960
      1 CONTINUE
      320 FORMAT(21X,17H*****CAUTION****/
      *21X,56HMORE THAN 10 DATA CARDS ENTERED--REMAINING CARDS IGNORED)  DAT02010
      GO TO 9999                   DAT02020
      999 IFLAG=4                  DAT02030
      9999 IF(IGEOSW.NE.1) GO TO 555  DAT02040
      DISKTM=1000,                  DAT02050
      DAT02060
      C*** CONVERT KM TO M,
      XT=PTS(1)*DISKTM            DAT02070
      YT=PTS(2)*DISKTM            DAT02080
      ZT=PTS(3)*DISKTM

```

```
X0=PTS(4)*DISKTM          DAT02090
Y0=PTS(5)*DISKTM          DAT02100
Z0=PTS(6)*DISKTM          DAT02110
555  RETURN               DAT02120
      END                  DAT02130
```

```

SUBROUTINE LT4M(H1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN, L4M00010
+MODEL,VIS,V11,V22,T1,ICLMAT,IERR,NR,IHAZE,MULDV) L4M00020
LOGICAL ISPOT,LOREAD,N16 L4M00030
COMMON /CONST/PI,PI2,CA,TWOP1,TORRMB,CDEGK L4M00040
COMMON /M01/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL L4M00050
COMMON /M02/W(34),RQ,TBOUND,JP,IM,ML,IP,JSTOR L4M00060
COMMON /M05/C1(501),C2(258),C3(86),C4(33),C5(6),C5DUM(9),C8(82), L4M00070
1 C11(4),C12(15),C14(21),C15(6) L4M00080
COMMON /M07/TR(67),FW(67),FD(67) L4M00090
COMMON /M08/SUM4,SUM5,SUM8,SUM11,SUM6 L4M00100
COMMON /M09/RADMAX,RADMIN,VRMAX,VRMIN L4M00110
COMMON /M03/F5(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FN02(9), L4M00120
1 O1(9),O2(9),PPMS02,PPMH3,PPMN02 L4M00130
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIR TU,NCLIM T,KSTOR,NPLOTUL L4M00140
COMMON /LOWEX/WPATH(68,16),WLAY(34,16),TB BY(68),TX(16),BETAEX, L4M00150
1 CLDHGT,NCLD L4M00160
COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1 L4M00170
COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL L4M00180
COMMON /BASPO T/ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16), L4M00190
1 BE(16),SINGWV,PF(65),LMAX L4M00200
COMMON /SPOTLO/ISPOT,LOREAD,N16 L4M00210
DIMENSION TRAN(16) L4M00220
DIMENSION RADA(16),RADG(16) L4M00230
C PLANCK RADIANCE FUNCTION L4M00240
FF(T,V)=1.190956E-16*(V**5)/(EXP(1.43879*V/T)-1.) L4M00250
C WATT CM-2 ST-1 MICRON-1 L4M00260
***** L4M00280
PROGRAM MODIFIED LOWTRAN CALCULATES THE TRANSMITTANCE L4M00290
OF THE ATMOSPHERE FROM 830 TO 1250, 2010 TO 3330, AND L4M00300
5010 TO 39990 CM-1 (0.25 TO 2.0, 3.0 TO 5.0, AND 8.0 TO 12.0 L4M00310
MICRONS) AT 20 CM-1 SPECTRAL INTERVALS ON A LINEAR WAVENUMBER L4M00320
SCALE. L4M00330
REFRACTION AND EARTH CURVATURE EFFECTS ARE EXCLUDED. ATMOSPHERE L4M00340
IS LAYERED IN ONE KM INTERVALS BETWEEN 0 AND 25 KM, 5 KM INTER- L4M00350
VALS FROM 25 TO 50 KM, A 20 KM LAYER FROM 50 TO 70 KM, A 30 KM L4M00360
LAYER FROM 70 TO 100 KM, AND ONE FROM 100 KM TO INFINITY. L4M00370
L4M00380
***** L4M00390
PROGRAM ACTIVATED BY SUBMISSION OF CARD SEQUENCE AS FOLLOWS L4M00400
***** L4M00410
CARD 1 MODEL,IHAZE,ITYPE,LEN,JP,NPLT,IM,ML,IEMISS, L4M00420
RO,TBOUND,BETAEX FORMAT (9I3,3F10.3) L4M00430
MODEL =0, METEOROLOGICAL DATA SPECIFIED L4M00440
=1, TROPICAL MODEL ATMOSPHERE L4M00450
=2, MIDLATITUDE SUMMER L4M00460
=3, MIDLATITUDE WINTER L4M00470
=4, SUBARCTIC SUMMER L4M00480
=5, SUBARCTIC WINTER L4M00490
=6, 1962 US STANDARD L4M00500
=7, NEW MODEL ATMOSPHERE L4M00510
=8, ISRAELI STANDARD ATMOSPHERE (YEAR, DAYTIME) L4M00520
=9, ISRAELI STANDARD ATMOSPHERE (YEAR, NIGHTTIME) L4M00530
** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK ** L4M00540
FOR SLANT PATHS IHAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES. L4M00550
IHAZE =0, NO AEROSOL ATTENUATION L4M00560
=1, MARITIME POLAR L4M00570
=2, MARITIME ARCTIC L4M00580
=3, CONTINENTAL POLAR L4M00590
=4, RAIN L4M00600
=5, SNOW L4M00610
=7, USER SUPPLIED EXTINCTION COEFFICIENT L4M00620
< READ ON ATM CARD - SEE CARD 3 BELOW > L4M00630
ITYPE =1, HORIZONTAL (CONSTANT PRESSURE) PATH L4M00640
=2, VERTICAL OR SLANT PATH BETWEEN 2 ALTITUDES L4M00650
=3, VERTICAL OR SLANT PATH TO SPACE L4M00660
LEN =0, NORMAL OPERATION L4M00670
=1, DOWNWARD LONG PATH L4M00680
JP =0, NORMAL OPERATION L4M00690
=1, SUPPRESS PRINT OF HORIZ AND VERTICAL PROFILES L4M00700
NPLT =0, NORMAL OPERATION L4M00710

```

=1, IN TRANSMISSION MODE WRITE, WAVELENGTH (UM), L4M00720
 H2O, CO₂, OZONE, N₂C, H₂O C, MOL SCAT, L4M00730
 NITRIC, SO₂, HNO₃, NO₂. IN EMISSION MODE, L4M00740
 WRITE WAVELENGTH (UM) AND RADIANCE PER MICRON, L4M00750
 RESULTS WILL BE WRITTEN ON NPLOTU (SEE COMMON BLOCK IOUNIT) L4M00760
 IM =1, RADIOSONDE DATA TO BE READ INITIALLY L4M00770
 =0, NORMAL OPERATION OR WHEN SUBSEQUENT CALCULATIONS L4M00780
 ARE TO BE RUN WITH MODEL = 7 L4M00790
 ML =NUMBER OF LEVELS TO BE READ IN FOR MODEL = 7 L4M00800
 ***IM AND ML ONLY USED WHEN MODEL = 7 AND THEN ONLY ON L4M00810
 FIRST CALCULATIONS WHEN DATA READ IN L4M00820
 IEMISS DETERMINES MODE OF EXECUTION OF PROGRAM L4M00830
 =0, TRANSMITTANCE MODE L4M00840
 =1, RADIANCE MODE L4M00850
 RO RADIUS OF THE EARTH (KM) AT LOCATION OF CALCULATION L4M00860
 ***DEFAULT WILL BE MIDLATITUDE VALUE OF 6371.23 KM WHEN L4M00870
 MODEL = 0 OR = 7 OTHERWISE DEFAULT IS EARTH RADIUS L4M00880
 FOR STANDARD MODEL ATMOSPHERE SPECIFIED BY MODEL L4M00890
 TBOUND TEMPERATURE OF EARTH (DEGREES K) AT LOCATION OF CALCUL4M00900
 ***USED ONLY IN RADIANCE MODE FOR SLANT PATHS WHICH INTERSECT EARTH L4M00910
 ***DEFAULT IS TEMPERATURE OF FIRST LAYER BOUNDARY TEMPERATURE L4M00920
 BETAEX USER SUPPLIED EXTINCTION COEFFICIENT, INPUT ONLY L4M00930
 WHEN IHAZE=7 L4M00940
 CARD 2 H1,H2,ANGLE,RANGE,BETA,VIS,CLDHGT FORMAT (7F10.3)
 H1 INITIAL ALTITUDE (KM) L4M00950
 H2 FINAL ALTITUDE (KM) L4M00960
 ANGLE INITIAL ZENITH ANGLE (DEG) L4M00970
 RANGE PATH LENGTH (KM) L4M00980
 BETA EARTH CENTER ANGLE SUBTENDED BY H1 AND H2 (DEG) L4M00990
 VIS SEA LEVEL VISUAL RANGE (KM) L4M01000
 CLDHGT HEIGHT OF BOTTOM OF CLOUD LAYER (KM), WHEN IHAZE NE 0L4M01010
 ***VIS NOT REQUIRED ON THIS CARD IF ICLMAT (EOMAIN) =1 OR L4M01020
 ***THIS IS FIRST LOOP THROUGH LT4 L4M01030
 ***SEE MANUAL FOR MORE DETAIL L4M01040
 L4M01050
 CARD 2A V1,V2,MULDY FORMAT (2F10.3,I2)
 V1 INITIAL FREQUENCY (CM**-1) L4M01060
 V2 FINAL FREQUENCY (CM**-1) L4M01070
 MULDY MULTIPLIER FOR FREQUENCY INCREMENT, WHERE THE L4M01080
 INCREMENT IS A MULTIPLE OF 20 (CM**-1). L4M01090
 OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY HR=1 IN EOMAIN) L4M01100
 CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2). L4M01110
 CARDS 2 - NUMBER OF VALUES: FORMAT (2(E10.4,1X)) L4M01120
 ONE VALUE OF WAVELENGTH (UM) AND RESPONSE FUNCTON PER CARD L4M01130
 CARD 3 IXY FORMAT (I3)
 IXY =0, EXIT LOWTRAN MODULE L4M01140
 =1, SELECT NEW WAVE FREQUENCY RANGE (CARD 2A) L4M01150
 =2, SELECT NEW DATA SEQUENCE (CARDS 1,2,2A,3) L4M01160
 =3, SELECT NEW CARD 2 AND CARD 3 L4M01170
 =4, SELECT NEW CARD 1 AND CARD 3 L4M01180
 ***FOR NON-STANDARD CONDITIONS SEE MANUAL L4M01190
 **** L4M01200
 V1=V11 L4M01210
 V2=V22 L4M01220
 KMAX=16 L4M01230
 ISPOT1=0 L4M01240
 RESPFN=0, L4M01250
 SUMRPF=0, L4M01260
 SUMINT=0, L4M01270
 IF (ISPOT) NPLT=0 L4M01280
 200 CONTINUE L4M01290
 IF (.NOT.LREAD) GO TO 400 L4M01300
 LREAD=.FALSE., L4M01310
 READ (IOUNIT,3300) IATM,NL L4M01320
 NL4=4*NL L4M01330
 DO 299 I=1,NL4 L4M01340
 299 READ (IOUNIT,3500) DUMMY L4M01350
 READ (IOUNIT,3510) PPMSO2,PPMNH3,PPMN02 L4M01360
 READ (IOUNIT,3700)(TR(I),FW(I),FO(I),I=1,67) L4M01370
 READ (IOUNIT,3800)(C1(I),I=1,501) L4M01380
 L4M01390
 L4M01400
 L4M01410

```

READ (LOUNIT,3800)(C2(I),I=1,258)          L4M01420
READ (LOUNIT,3800)(C3(I),I=1,86)           L4M01430
READ (LOUNIT,3900)(C4(I),I=1,33)           L4M01440
READ(LOUNIT,3810)(C5(I),I=1,6)             L4M01450
READ (LOUNIT,3900)(C8(I),I=1,82)           L4M01460
READ (LOUNIT,4000)(C1)(I),I=1,4)           L4M01470
READ (LOUNIT,4010)(C12(I),I=1,15)          L4M01480
READ (LOUNIT,4020)(C14(I),I=1,21)          L4M01490
READ (LOUNIT,4020)(C15(I),I=1,6)           L4M01500
READ (LOUNIT,4010)(FS(I),S1(I),S2(I),I=1,9) L4M01510
READ (LOUNIT,4010)(FNH3(I),FH1(I),FH2(I),I=1,9) L4M01520
READ (LOUNIT,4010)(FNH2(I),O1(I),O2(I),I=1,9) L4M01530
REWIND LOUNIT                                L4M01540
400 IF (.ISPOT) GO TO 710                    L4M01550
CALL CKER (V1,V2,DV,IV1,IV2,IERR,MULDV,ISPOT,TRANC1) L4M01560
IF (IERR.EQ.1) RETURN                         L4M01570
JP=0                                         L4M01580
JSTUR=0                                      L4M01590
IP=0                                         L4M01600
IF (.NOT.ISPOT) GO TO 700                    L4M01610
BETA=0                                       L4M01620
RANGE=0                                      L4M01630
RO=0                                         L4M01640
IF (ITYPE.EQ.1) RANGE=ANGLE                 L4M01650
IF (IXY.EQ.0) GO TO 700                     L4M01660
401 GO TO (500,700,600,680),IXY              L4M01670
500 AVW=1,E+04/V1                           L4M01680
ALAM=1,E+04/V2                           L4M01690
SUMA=0                                      L4M01700
GO TO 1100                                 L4M01710
600 IF (MODEL.EQ.0) GO TO 800               L4M01720
ISPO1=1                                     L4M01730
GO TO 1000                                 L4M01740
700 CONTINUE                                L4M01750
RE=6371.23                                  L4M01760
680 IF (.NOT.ISPOT) READ (IOIN,3300) MODEL,IHAZE,ITYPE, L4M01770
    1 LEN,JP,NPLT,IM,ML,IEMISS,RO,TBOUND      L4M01780
C***** IEMISS=0=TRANSMISSION MODE / IEMISS=1=EMISSION MODE L4M01790
    IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,4100) L4M01800
    IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4200) L4M01810
    IF (ISPOT) GO TO 800                     L4M01820
    IF (MODEL.EQ.0.OR.MODEL.EQ.7) GO TO 210   L4M01830
710 READ(LOUNIT,3300)IATM,NL                L4M01840
MSKIP=MODEL-1/2                            L4M01850
IF(MSKIP.EQ.0) GO TO 220                  L4M01860
IF (MSKIP.EQ.4) MSKIP=3                   L4M01870
DO 230 J=1,MSKIP                          L4M01880
DO 230 I=1,NL                            L4M01890
230 READ(LOUNIT,3500) DUMMY               L4M01900
220 CONTINUE                                L4M01910
C ISRAELI STD ATM READS
    IF (MODEL.EQ.8) GO TO 270               L4M01920
    IF (MODEL.EQ.9) GO TO 250               L4M01930
    IF (<2*MODEL/2>.EQ.MODEL) GO TO 250   L4M01940
270 DO 240 I=1,NL                          L4M01950
240 READ(LOUNIT,3500)Z(I),P(I),T(I),WAC(I),WH(I),WD(I) L4M01960
    GO TO 210                               L4M01970
250 DO 260 I=1,NL                          L4M01980
260 READ(LOUNIT,3550)Z(I),P(I),T(I),WAC(I),WH(I),WD(I) L4M02000
210 REWIND LOUNIT                           L4M02010
    IF (.ISPOT) RETURN                      L4M02020
800 M=MODEL                                L4M02030
900 IF (RO.GT.0) RE=RO                      L4M02040
    LENTOR=LEN                                L4M02050
1000 CALL ABSORB(IXY,IERR,W,V1,V2,DV,SUMA,MULDV,ANGLE,LEN,ITYPE,H1,H2, L4M02060
    1 MODEL,ISPO1,RANGE,BETA,VIS,ICLMAT,IV1,IV2,IVD)      L4M02070
    IF (IERR.EQ.1) TRANC1=1                  L4M02080
    IF (IERR.EQ.1) RETURN                   L4M02090
1100 CONTINUE                                L4M02100
    IF (.NOT.ISPOT) WRITE (IOOUT,4300)       L4M02110

```

```

IF (.NOT.ISPOT) WRITE(I00UT,4400) (W(I),I=1,6),W(8),W(10) L4M02120
IF (.NOT.ISPOT) WRITE(I00UT,4401) L4M02130
IF (.NOT.ISPOT) WRITE(I00UT,4500) W(11),W(12),W(13),W(14), L4M02140
* W(15),W(16) L4M02150
1200 CONTINUE L4M02160
C NCLD WILL BE THE INDEX OF THE LAYER ABOVE CLDHGT L4M02170
DO 5 IC LD=2,6 L4M02180
NCLD=ICLD L4M02190
IF (CLDHGT.LT.Z(ICLD)) GO TO 6 L4M02200
CONTINUE L4M02210
CONTINUE L4M02220
5 IF (CLDHGT.GT.Z(6)) WRITE(I00UT,7) L4M02230
6 FORMAT(1H,25H**** WARNING FROM LOWTRAN,/,1X,14H CLOUD BASE IS , L4M02240
7 1 23HLIMITED TO 4 KM MAXIMUM/)
I=1 L4M02250
L=1 L4M02260
IV=IV1 L4M02270
ICOUNT=0 L4M02280
IF (N16) KWAVE=0 L4M02290
IF (IEMISS.EQ.0) GO TO 1300 L4M02300
RADSUM=0.0 L4M02310
FACTOR=0.5 L4M02320
CALL LTPATH(WLAY,WPATH,TBBY,ANGLE,LEN,ITYPE,H1,H2,MODEL) L4M02330
IF (.NOT.ISPOT) WRITE(I00UT,4600) L4M02340
IF (.NOT.ISPOT) WRITE(I00UT,4700) L4M02350
L4M02360
C***** BEGINNING OF TRANSMITTANCE CALCULATIONS L4M02370
1300 CONTINUE L4M02380
IF (N16) KWAVE=KWAVE+1 L4M02390
SUMV=0. L4M02400
TL0LD=1. L4M02410
TSOLD=1. L4M02420
TX7=1. L4M02430
TX10=1. L4M02440
IKLO=1. L4M02450
TOL0=1. L4M02460
L4M02470
C IF (IEMISS.EQ.0) IKMAX=IKLO L4M02480
ONLY ONE LOOP FOR TRANSMISSION: LOOP OVER LAYERS FOR EMISSION L4M02490
DO 2300 IK=IKLO,IKMAX L4M02500
L4M02510
C IF (IEMISS.EQ.0) GO TO 1500 L4M02520
C TRANSFER CUMULATIVE ABSORBER AMOUNTS FOR TH IK TH LEVEL AND L4M02530
THE K TH ABSORBER - EMISSION ONLY. L4M02540
DO 1400 K=1,KMAX L4M02550
W(K)=WPATH(IK,K)
1400 CONTINUE L4M02560
1500 IJ=IK L4M02570
IF (ICOUNT.EQ.0) GO TO 1600 L4M02580
IF (ICOUNT.EQ.50) GO TO 1600 L4M02590
GO TO 1700 L4M02600
1600 ICOUNT=0 L4M02610
IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE(I00UT,4800) L4M02620
1700 DO 1800 K=1,KMAX L4M02630
TX(K)=1.0 L4M02640
1800 CONTINUE L4M02650
ICOUNT=ICOUNT+1 L4M02660
V=FLOAT(IV) L4M02670
ALAM=1.E+04/V L4M02680
I=(IV-830)/20+1 L4M02690
SUM4=0. L4M02700
SUM5=0. L4M02710
SUM6=0. L4M02720
SUM8=0. L4M02730
SUM11=0. L4M02740
CALL FREQL(I,IV,W,TX) L4M02750
TX(9)=SUM4+SUM5+SUM8+SUM11+SUM6 L4M02760
IF (TX(9).EQ.0.0) GO TO 2000 L4M02770
IF (TX(9).LE.0.1) GO TO 1900 L4M02780
IF (TX(9).GT.2.0) GO TO 2100 L4M02790
TX(9)=EXP(-TX(9)) L4M02800
GO TO 2200 L4M02810
1900 TX(9)=1.0-TX(9)+0.5*TX(9)*TX(9)

```

```

2000 GO TO 2200 L4M02820
    TX(9)=1.0 L4M02830
    GO TO 2200 L4M02840
2100 TX(9)=0 L4M02850
2200 TX(9)=TX(1)*TX(2)*TX(3)*TX(9)*TX(12)*TX(13)*TX(14) L4M02860
C     AEROSOL COMPUTATIONS UNTIL LABEL 1 L4M02870
C     IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 1 L4M02880
C
C     IF SPOT IS CALLING LT4M WITH ITYPE = 3, DO NOT INCLUDE AEROSOLS L4M02890
C
C     IF (ISPOT.AND.ITYPE.EQ.3) GO TO 1 L4M02900
C     EXT55=3.914/VIS L4M02910
C     UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE L4M02920
C     PASS HORIZONTAL DIST IF ITYPE=1, SLANT DISTANCE IF ITYPE GT 1. L4M02930
C     IF (ITYPE.EQ.1) RNG=RANGE L4M02940
C     IF (ITYPE.EQ.2.AND.(H2.GT.H1).AND.(RANGE.GT..5/COS(ANGLE*CA))) L4M02950
C     1  RNG=.5/COS(ANGLE*CA) L4M02960
C     IF (ITYPE.EQ.2.AND.(H2.LT.H1).AND.,(RANGE.GT..5/COS(ANGLE*CA))) L4M02970
C     1  RNG=.5/COS((180.-ANGLE)*CA) L4M02980
C     2  RNG=.5/COS((180.-ANGLE)*CA) L4M02990
C     IF (ITYPE.EQ.3.AND.(RANGE.GT..5/COS(ANGLE*CA))) L4M03000
C     1  RNG=.5/COS(ANGLE*CA) L4M03010
C     IF (ITYPE.EQ.3.AND.RANGE.LT..0001) L4M03020
C     1  RNG=.5/COS(ANGLE*CA) L4M03030
C     ISLANT=ITYPE-1 L4M03040
C     CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL L4M03050
C     CALL XSCALE(ALAM,88.,EXT55,TX7,IERR,ISLANT,IHAZE,RNG,ANGLE) L4M03060
C     IF (IERR.EQ.1) RETURN L4M03070
C     USER OPTIONS L4M03080
C     IF (IHAZE.EQ.7) TX7=EXP(-BETAEX*RANGE) L4M03090
C     IF (ISPOT.AND.IHAZE.EQ.8) TX7=EXP(-BECKWAVE)*RANGE) L4M03100
C     CONTINUE L4M03110
C     TX(9)=TX(9)*TX7 L4M03120
C     IF (IV.GE.13000) TX(3)=TX(8) L4M03130
C     TNEW=TX(9) L4M03140
C     IF (IEMISS.EQ.0) GO TO 2500 L4M03150
C     COMPUTER LIMITS L4M03160
C     BBIK=0.0 L4M03170
C     IF (ABS(1.43879*V/TBBY(IK)).LT.85.) BBIK=FF(TBBY(IK),V) L4M03180
C     AEROSOL COMPUTATIONS UNTIL LABEL 2 L4M03190
C     IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 2 L4M03200
C     FIND AEROSOL ABSORPTION IN DIFFERENT WAVELENGTH BANDS FROM EXTN L4M03210
C     IF (ALAM.LT.2.) TX10=1. L4M03220
C     IF (ALAM.GE.3..AND.ALAM.LE.5.) TX10=TX7**.2 L4M03230
C     IF (ALAM.GE.8..AND.ALAM.LE.12.05) TX10=TX7**.45 L4M03240
C     IF (ALAM.GE.12.05) TX10=TX7**.45 L4M03250
C     CONTINUE L4M03260
C     TLNEW=TX(9)*TX10/(TX(6)*TX7) L4M03270
C     TSNEW=TX7*TX6)/TX10 L4M03280
C     DTAU=ABS(TLOLD-TLNEW) L4M03290
C     IF (DTAU.LT.1.0E-5.AND.TLNEW.LT.1.0E-5) GO TO 2400 L4M03300
C     SUMV=SUMV+(TOLD-TNEW)*BBIK L4M03310
C     TLOLD=TLNEW L4M03320
C     TSOLD=TSNEW L4M03330
C     TOLD=TNEW L4M03340
C     CONTINUE L4M03350
C     TAUG=0.0 L4M03360
C     IF (HM11.LE.0.0.AND.IL.EQ.1) TAUG=TX(9) L4M03370
C     T1=T(1) L4M03380
C     IF (TBOUND.GT.0.0) T1=TBOUND L4M03390
C     COMPUTER LIMITS L4M03400
C     BBC=0.0 L4M03410
C     IF (ABS(1.43879*V/T1).LT.85.) BBC=FF(T1,V)*TAUG L4M03420
C     IF (N16) RADG(KWAVE)=BBC*1.E+04 L4M03430
C     IF (N16) RADA(KWAVE)=SUMV*1.E+04 L4M03440
C     IF (HM11.LE.0) SUMV=SUMV+BBC L4M03450
C     SUMVV=SUMV L4M03460
C     IF (IV.GT.IV1) FACTOR=1.0 L4M03470
C     IF (IV.GE.IV2) FACTOR=0.5 L4M03480
C     SUMV=(1.0E+04/V**2)*SUMV L4M03490
C
C     SUMV=(1.0E+04/V**2)*SUMV L4M03500
C
C     SUMV=(1.0E+04/V**2)*SUMV L4M03510

```

```

RADSUM=RADSUM+DV*FACTOR*SUMV          L4M03520
IF (.NOT.ISPOT) WRITE (I00UT,4900) V,    L4M03530
1   ALAM,SUMV,SUMVV,RADSUM,TX(9),TX7,TX10  L4M03540
IF (NPLT.EQ.1) WRITE (NPLOTU,98) ALAM,SUMVV  L4M03550
FORMAT (F7.4,1X,E13.5)                 L4M03560
IF (SUMV.GE.RADMAX) VRMAX=V           L4M03570
IF (SUMV.GE.RADMAX) RADMAX=SUMV       L4M03580
IF (SUMV.LE.RADMIN) VRMIN=V          L4M03590
IF (SUMV.LE.RADMIN) RADMIN=SUMV       L4M03600
2500 AB=1.-TX(9)                      L4M03610
IF (IV.EQ.IV1.OR.IV.GE.IV2) AB=0.5*AB  L4M03620
SUMA=SUMA+AB*Dv                      L4M03630
IF (IEMISS.EQ.1) GO TO 2600          L4M03640
IF (.NOT.ISPOT) WRITE (I00UT,5000) IV,  L4M03650
1   ALAM,(TX(K),K=1,6),TX(11),TX(12),TX(13),TX(14),  L4M03660
2   SUMA,TX(9),TX7                   L4M03670
IF (NPLT.EQ.1) WRITE (NPLOTU,99) ALAM,(TX(K),K=1,6),(TX(J),J=11,14) L4M03680
99 FORMAT (F7.4,10(1X,F6.4))          L4M03690
2600 CONTINUE
RESPFN=RESFH(NR,ALAM)                L4M03700
SUMRPF=SUMRPF+RESPFN                L4M03710
IF (IV.GT.IV1) SUMINT=SUMINT+.5*(OLDTX9*OLDRFN+TX(9)*RESPFN)*  L4M03720
+ABSC1./FLOAT(IV)-1./FLOAT(IV-IDV))*1.E+04  L4M03730
IF (K16) TRAN(KWAVE)=TX(9)          L4M03740
OLDTX9=TX(9)                        L4M03750
OLDRFN=RESPFN                      L4M03760
IV=IV+IDV                           L4M03770
IF (IV.GE.IV2) GO TO 2700          L4M03780
GO TO 1300                           L4M03790
2700 CONTINUE
IF (NR.NE.1) SUMRPF=1.               L4M03800
SUMINT=SUMINT/(SUMRPF*1.E+04*ABSC1./FLOAT(IV1)-1./FLOAT(IV2)))  L4M03810
IF (.NOT.ISPOT) TRAN(1)=SUMINT      L4M03820
IF (.NOT.ISPOT.AND.NR.EQ.1) WRITE (I00UT,3250) SUMINT  L4M03830
IF (.NOT.ISPOT.AND.NR.NE.1) WRITE (I00UT,3275) SUMINT  L4M03840
RESPFN=0.                            L4M03850
SUMRPF=0.                            L4M03860
SUMINT=0.                            L4M03870
IF ((IEMISS.EQ.1).AND.(<.NOT.ISPOT)) WRITE (I00UT,5100)  L4M03880
1   VRMIN,RADMIN,VRMAX,RADMAX      L4M03890
JSTOR=0.                            L4M03900
AB=1.0-SUMA/FLOAT(IV-IV1)          L4M03910
IF (ISPOT) RETURN                  L4M03920
WRITE (I00UT,5200) IV1,IV,SUMA,AB  L4M03930
IF ((IEMISS.EQ.1).AND.(<.NOT.ISPOT)) WRITE (I00UT,5300) RADSUM  L4M03940
IF (.NOT.ISPOT) READ (I0IN,3300) IXY  L4M03950
IF (IXY.EQ.0) GO TO 3100          L4M03960
IF (IXY.EQ.0) GO TO 3100          L4M03970
GO TO (2800,700,2900,680,3100),IXY  L4M03980
2800 CONTINUE
READ (I0IN,5400) V1,V2,MULDV      L4M03990
CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(1))  L4M04000
IF (IERR.EQ.1) RETURN            L4M04010
AVW=10000./V1                     L4M04020
ALAM=10000./V2                     L4M04030
WRITE (I00UT,5500) V1,V2,DV,ALAM,AVW  L4M04040
SUMA=0.0                           L4M04050
GO TO 1100                         L4M04060
2900 IF (MODEL.EQ.0) GO TO 800     L4M04070
GO TO 401                         L4M04080
3000 CONTINUE
READ (I0IN,3300) MODEL,IHAZE,ITYPE,LEN,JP,NPLT,IM,  L4M04090
1   ML,IEMISS,RO,TBOUND,BETAEX      L4M04100
IF (IEMISS.EQ.1) WRITE (I00UT,4100)  L4M04110
IF (IEMISS.EQ.0) WRITE (I00UT,4200)  L4M04120
LENTOR=LEN                         L4M04130
GO TO 800                          L4M04140
3100 RETURN                         L4M04150
C
3250 FORMAT (/,1X,48HWAVELENGTH AND SENSOR INTEGRATED TRANSMISSION =  L4M04160
+,E10.4)                           L4M04170
                                         L4M04180
                                         L4M04190
                                         L4M04200
                                         L4M04210

```

```

3275 FORMAT (/,1X,37HWAVELENGTH INTEGRATED TRANSMISSION = ,E10.4) L4M04220
3300 FORMAT (9I3,3F10.3) L4M04230
3500 FORMAT (F6.1,2(E9.3,F5.1,E9.3,2E7.1)) L4M04240
3510 FORMAT(3F8.3) L4M04250
3550 FORMAT(F6.1,37X,E9.3,F5.1,E9.3,2E7.1) L4M04260
3700 FORMAT (4(F6.3,2F7.4)) L4M04270
3800 FORMAT (15F5.2) L4M04280
3810 FORMAT(15F5.3) L4M04290
3900 FORMAT (8E9.2) L4M04300
4000 FORMAT (12F6.3) L4M04310
4010 FORMAT(10F8.4) L4M04320
4020 FORMAT(10F8.3) L4M04330
4100 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,//,1X, L4M04340
1 45HPROGRAM WILL BE EXECUTED IN THE EMISSION MODE) L4M04350
4200 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,//,1X, L4M04360
1 49HPROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE) L4M04370
4300 FORMAT (/10X,38H EQUIVALENT SEA LEVEL ABSORBER AMOUNTS L4M04380
1//21X,56HWATER VAPOUR CO2 ETC. OZONE NITROGEN (CONT), L4M04390
2 42H H2O (CONT) MOL SCAT OZONE(U-V)/24X, L4M04400
3 7HGM CM-2,10X,2HKM,10X,6HATM CM,10X,2HKM,9X,7HGM CM-2, L4M04410
4 10X,2HKM,10X,6HATM CM) L4M04420
4400 FORMAT(/10X,10H W(1-6,8)=,7(E14.3)/,1X,10X,7H W(10)=,58X,E14.3/) L4M04430
4401 FORMAT(/23X,11HNITRIC ACID,8X,3HSO2,11X,3HNN3,11X,3HN02/) L4M04440
4500 FORMAT(/10X,10H W(11-16)=,6(E14.3)/) L4M04450
4600 FORMAT (1H1,30X,28HRADIANCE(WATTS/CM2-STER-XXX)) L4M04460
4700 FORMAT (1H,10X,37HFRC(CM-1) WVL(MICRON) PER CM-1 L4M04470
1 0HPER MICRON,26H INTEGRAL TRANS,1X,4(1H-), L4M04480
2 1H AERO TRAN,4(1H-),/,1X,84X,17H EXTN ABS) L4M04490
4800 FORMAT (1H1,/,1X,2X,15HFREQ WAVELENGTH,2X,3HH20,3X,4HC02+,4X, L4M04500
1 30HOZONE N2 C H2O C MOL S,1X, L4M04510
2 22HNITRIC SO2 HN03,4X,16HN02 INTEGRATED, L4M04520
3 2X,13HTOTAL AEROSOL/1X,1X,13H CM-1 MICRONS,10(3X,5HTRANS), L4M04530
4 2X,24HABSORPTION TRANS TRANS) L4M04540
4900 FORMAT (1H,10X,F8.1,F13.6,3E13.5,F13.6,1X,F7.5,3X,F7.5) L4M04550
5000 FORMAT (1H,16,11F8.4,F11.4,F8.4,1X,F7.5) L4M04560
5100 FORMAT (1H0,8H RADMIN ,F12.3,E12.5,/,8H RADMAX ,F12.3, L4M04570
1 E12.5) L4M04580
5200 FORMAT (1H0,26H INTEGRATED ASORPTION FROM,15,4H TO ,15, L4M04590
1 7H CM-1 =,F10.2,25H, AVERAGE TRANSMITTANCE =,F6.4) L4M04600
5300 FORMAT (1H,22H INTEGRATED RADIANCE =,E12.5,13H WATT CM -2 ,2HSR) L4M04610
5400 FORMAT (2F10.3,I2) L4M04620
5500 FORMAT (/10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO , L4M04630
1 4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1 < L4M04640
2 ,F6.2,3H -,F5.2,10H MICRONS )) L4M04650
END L4M04660

```

```

SUBROUTINE ABSORB(IXY,IERR,W,V1,V2,DV,SUMA,MULDV,ANGLE,LEN,ITYPE, ABS00010
1 H1,H2,MODEL,ISPOT1,RANGE,BETA,VIS,ICLMAT,IV1,IV2,IDX) ABS00020
COMMON /GEOMET/PTS(15),IGEOSW ABS00030
COMMON /CLYMAT/TEMP,PRESS,RH1,AM1,DP1,VIS1,CLDMAT, ABS00040
1 CLDHYT,FOGPRB,WNDYEL,WDDIR,IPASCT ABS00050
COMMON /CONST/PI,PI2,CA,TWOP1,TORRMB,CDECK ABS00060
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NFLOTUABS00070
COMMON /M01/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL ABS00080
COMMON /M02/WC(34),RO,TBOUND,JP,IM,ML,IP,JSTOR ABS00090
COMMON /M09/RADMAX,RADMIN,VRMAX,VRMIN ABS00100
COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1 ABS00110
COMMON /LQWEX/WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX, ABS00120
1 CLDHGT,NCLD ABS00130
COMMON /SPOTLD/ISPOT,LOREAD,H16 ABS00140
COMMON /M03/FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FN02(9), ABS00150
1 S1(9),S2(9),PPMS02,PPMNH3,PPMN02 ABS00160
LOGICAL ISPOT,H16,LOREAD ABS00170
DIMENSION VH(16),WC(16),E(16) ABS00180
EH(?,?) REPLACES HSTOR ABS00190
EH(?,?) REPLACES HMIX ABS00200
DATA EH(9,?),I=1,34)/9*0.,0.1,0.33,0.8,1.2,1.4,1.6,1.8,1.9,
1 2.0,2.1,2.3,3.0,3.7,4.2,5.2,6.0,3.8,2.6,0.22,6*0.0? ABS00210
F(A)=EXP(18.9766-14.9595*A-2.43882*A*A)*A ABS00220
TMPVIS=VIS ABS00230
IF (ISPOT1.EQ.1) GO TO 200 ABS00240
IF (MODEL.EQ.0) GO TO 400 ABS00250
IF (IXY.EQ.3) GO TO 100 ABS00260
IF (M.EQ.7.AND.IM.NE.0) GO TO 400 ABS00270
IF (IXY.GT.3) GO TO 1500 ABS00280
WHEN IXY=0 VIS IS READ IN MAIN ABS00290
100 IF (.NOT.ISPOT) READ (I0IN,6200) H1,H2,ANGLE,RANGE, ABS00300
1 BETA,VIS,CLDHGT ABS00310
IF (IGEOSW.NE.1) GO TO 111 ABS00320
H1=PTS(3) ABS00330
H2=PTS(6) ABS00340
RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+ ABS00350
1 +(PTS(3)-PTS(6))**2) ABS00360
111 CONTINUE ABS00370
IF (IXY.EQ.0) VIS=TMPVIS ABS00380
200 X1=RE+H1 ABS00390
X2=RE+H2 ABS00400
IF (ITYPE.EQ.3) GO TO 1000 ABS00410
IF (ITYPE.EQ.1) GO TO 1500 ABS00420
IF (RANGE.EQ.0.) GO TO 1200 ABS00430
IF (.NOT.ISPOT) WRITE (I0OUT,6300) H1,H2,ANGLE,RANGE, ABS00440
1 BETA,VIS ABS00450
IF (H2.EQ.0.AND.ANGLE.NE.0) GO TO 300 ABS00460
ANGLE= ACOS(0.5*((H2-H1)*(1.+X2/X1)/RANGE-RANGE/X1))/CA ABS00470
GO TO 1400 ABS00480
300 X2=SQRT((X1/RANGE+RANGE/X1+2.0*COS(ANGLE*CA))*X1* ABS00490
1 RANGE) ABS00500
H2=X2-RE ABS00510
GO TO 1400 ABS00520
400 CONTINUE ABS00530
IF (ML.LE.0) ML=1 ABS00540
DO 900 K=1,ML ABS00550
CLIMATE OPTION - SEE COMMON /CLYMAT/
1 IF (M.EQ.0.AND..NOT.ISPOT) READ (I0IN,6400) H1,P(1), ABS00560
1 TMP,DP,RH,WH(K),WO(K),VIS,RANGE ABS00570
1 IF (IGEOSW.NE.1) GO TO 444 ABS00580
H1=PTS(3) ABS00590
H2=PTS(6) ABS00600
RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+ ABS00610
1 +(PTS(3)-PTS(6))**2) ABS00620
444 CONTINUE ABS00630
IF (ICLMAT.NE.1) GO TO 500 ABS00640
TMP=TEMP ABS00650
P(1)=PRESS ABS00660
DP=DP1 ABS00670
RH=RH1 ABS00680

```

```

WH(K)=AH1
VIS=VIS1
500 IF ((IXY.EQ.0.AND.ICLMAT.NE.1)) VIS=TMPVIS
    IF ((M.GT.0).AND.(<.NOT.ISPOT)) READ (IOIN,6400) Z(K),
       P(K),TMP,DP,RH,WH(K),WOK(K)
    IF ((M.EQ.0).AND.(<.NOT.ISPOT)) WRITE (IOOUT,6500) H1,
       P(K),TMP,DP,RH,WH(K),WOK(K),VIS,RANGE
    IF ((M.EQ.0)) Z(K)=H1
    J=IFIX(Z(K)+1.0E-6)+1.
    IF (Z(K).GE.25.) J=(Z(K)-25.)/5.0+26.
    IF (Z(K).GE.50.0) J=(Z(K)-50.)/20.+31.
    IF (Z(K).GE.70.) J=(Z(K)-70.)/30.+32.
    IF (J.GT.33) J=33
    FAC=Z(K)-FLOAT(J-1)
    IF (J.LT.26) GO TO 600
    FAC=(Z(K)-5.0*FLOAT(J-26)-25.)/5.
    IF (J.GE.31) FAC=(Z(K)-50.)/20.
    IF (J.GE.32) FAC=(Z(K)-70.)/30.
    IF (FAC.GT.1.0) FAC=1.0
600 L=J+1
    T(K)=TMP+CDEGK
    TT=CDEGK/T(K)
    IF (RH.LE.0.0) TT=CDEGK/(CDEGK+DP)
    IF (WH(K).LE.0.0) WH(K)=F(TT)
    IF (RH.GT.0.0) WH(K)=0.01*RH*WH(K)
    EH(7,K)=0.0
    IF (EH(9,J).LE.0.) GO TO 700
    EH(7,K)=EH(9,J)*(EH(9,L)/EH(9,J))*FAC
700 CONTINUE
    IF (MODEL.EQ.0) GO TO 1500
    IF ((K.EQ.1).AND.(<.NOT.ISPOT)) WRITE (IOOUT,6600)
    IF (<.NOT.ISPOT) WRITE (IOOUT,6400) Z(K),P(K),TMP,DP,
       RH,WH(K),WOK(K)
900 CONTINUE
    IM=0
    NL=ML
C NOTE THAT Z(I) MAY NOT CORRESPOND TO THE VALUES GIVEN FOR STANDARD ABS01070
C MODEL ATMOSPHERES ABS01080
    IF (IXY.GE.3) GO TO 1500
    GO TO 100
1000 IF (RANGE.GT.0.0) GO TO 1100
    GO TO 1500
1100 ITYPE=2
    BETA= ACOS(0.5*(RANGE*RANGE/(X1*X2)-X2/X1-X1/X2))/CA
1200 IF (BETA.EQ.0.0) GO TO 1300
    BET=CA*BETA
    X2=RE+H2
    ANGLE=ATAN(X2*SIN(BET)/(X2*COS(BET)-X1))/CA
    IF (ANGLE.LT.0.) ANGLE=ANGLE+PI
    RANGE=X2*SIN(BET)/SIN(ANGLE*CA)
    BET=BETA
    GO TO 1500
1300 RANGE=(X2/X1)**2-(SIN(ANGLE*CA))**2
    IF (RANGE.GE.0.0) RANGE=X1*(SQRT(RANGE)-ABS(COS(ANGLE*CA)))
1400 IF (ANGLE.NE.0.0.OR.ANGLE.NE.180.) BET=ASIN(RANGE*SIN(ANGLE*CA)/X2)/X2
    IF (ANGLE.LT.0.) ANGLE=ANGLE+PI
    IF (RANGE.LT.0.0) RANGE=-RANGE
    BET=BET/CA
    IF (<.NOT.ISPOT) WRITE (IOOUT,6300) H1,H2,ANGLE,RANGE.
    1   BET,VIS
1500 CONTINUE
    DO 1600 I=1,NL
    DO 1600 J=1,KMAX
1600 WLAY(I,J)=0.
    SUMA=0.
C WHEN IXY=0 V1,V2,MULDV ARE READ IN EOSAEL.MAIN ABS01360
    IF ((IXY.EQ.1.OR.IXY.EQ.2).AND.(<.NOT.ISPOT))
    1   READ (IOIN,6250) V1,V2,MULDV ABS01370
    IF ((IXY.EQ.1.OR.IXY.EQ.2).AND.(<.NOT.ISPOT))
    + CALL CKER (V1,V2,DV,IV1,IV2,IV3,IERR,MULDV,ISF,T,TMPVIS) ABS01380
    ABS01390
    ABS01400

```

```

IF (IERR.EQ.1) RETURN
IF ((ITYPE.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,6700) ABS01410
1   H1,RANGE ABS01420
IF ((ITYPE.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,6800) ABS01430
1   H1,H2,ANGLE ABS01440
IF ((ITYPE.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,6900) ABS01450
1   H1,ANGLE ABS01460
IF (MODEL.EQ.0),M=7 ABS01470
IF ((M.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,7200) M ABS01480
IF ((M.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,7300) M ABS01490
IF ((M.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,7400) M ABS01500
IF ((M.EQ.4).AND.(.NOT.ISPOT)) WRITE (IOOUT,7500) M ABS01510
IF ((M.EQ.5).AND.(.NOT.ISPOT)) WRITE (IOOUT,7600) M ABS01520
IF ((M.EQ.6).AND.(.NOT.ISPOT)) WRITE (IOOUT,7700) M ABS01530
IF ((M.EQ.8).AND.(.NOT.ISPOT)) WRITE (IOOUT,7800) M ABS01540
IF ((M.EQ.9).AND.(.NOT.ISPOT)) WRITE (IOOUT,7900) M ABS01550
AVW=10000./V1 ABS01560
ALAM=10000./V2 ABS01570
RADMIN=1.0E+38 ABS01580
RADMAX=0. ABS01590
VRMIN=0. ABS01600
VRMAX=0. ABS01610
IF (.NOT.ISPOT) WRITE (IOOUT,8000) V1,V2,DV,ALAM,AVW ABS01620
AVW=0.5E-4*(V1+V2) ABS01630
AVW=AVW*AVW ABS01640
IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8100) ABS01650
IF (ITYPE.EQ.1) GO TO 2100 ABS01660
DO 1800 K=1,KMAX ABS01670
VHK(K)=0.0 ABS01680
1800 CONTINUE ABS01690
BETA=0.0 ABS01700
SR=0.0 ABS01710
IP=0 ABS01720
ABS01730
***** NOW DEFINE CONSTANT PRESSURE PATH QUANTITES EH(1-8)
Y=CA*ANGLE ABS01740
SPHI=SIN(Y) ABS01750
R1=(RE+H1)*SPHI ABS01760
IF (H1.GT.Z(NL)) GO TO 1900 ABS01770
GO TO 2100 ABS01780
1900 X=(RE+Z(NL))/(RE+H1) ABS01790
IF (SPHI.GT.X) GO TO 2000 ABS01800
H1=Z(NL) ABS01810
J1=NL ABS01820
SPHI=SPHI/X ABS01830
ANGLE=180.0-ASIN(SPHI)/CA ABS01840
R1=(RE+H1)*SPHI ABS01850
GO TO 2100 ABS01860
2000 HMIN=R1-RE ABS01870
IF (.NOT.ISPOT) WRITE (IOOUT,8200) HMIN ABS01880
GO TO 6000 ABS01890
2100 DO 2400 I=1,NL ABS01900
PS=P(I)/1013.0 ABS01910
TS=CDEGK/T(I) ABS01920
X=PS*TS ABS01930
ABS01940
C--- COMPUTE MASS DENSITY (G M-3) FROM IDEAL GAS LAW ---
C--- 1292.02 = DENSITY OF STANDARD COMPOSITION AIR AT STP ---
1292.02 = 1292.02*X ABS01950
WA(I) = 1292.02*X ABS01960
PT=PS*SQRT(TS) ABS01970
D=0.1*WH(I) ABS01980
EH<1,I>=D*PT**0.9 ABS01990
EH<2,I>=X*PT**0.75 ABS02000
EH<4,I>=0.8*PT*X ABS02010
PPW=4.56E-5*D*CDEGK/TS ABS02020
TS1=(296.0/CDEGK)*TS ABS02030
EH<5,I>=D*PPW*EXP(6.08*(TS1-1.0))+0.002*D*(PS-PPW) ABS02040
EH<10,I>=D*(PPW+0.12*(PS-PPW))*EXP(4.56*(TS1-1.0)) ABS02050
EH<6,I>=X ABS02060
EH<8,I>=46.6667*W(I) ABS02070
EH<3,I>=EH<8,I>*PT**0.4 ABS02080
C EH<11,I>=HNO3 ABSORBER AMOUNT (ATM-CM)/KM ABS02090
ABS02100

```

```

EH(11,I)=PS*TS*EH(9,I)*1.0E-04          ABS02110
IF (MODEL.EQ.0.OR.MODEL.EQ.7) EH(11,I)=PS*TS*EH(7,I)*1.0E-04      ABS02120
*****EH(12,I)=S02 ABSORBER AMOUNT (ATM-CM)/KM      ABS02130
EH(12,I)=0.772E-04*PPMS02*WA(I)*PS**0.07122*TS**0.06159      ABS02140
EH(13,I)=0.772E-04*PPMNH3*WA(I)*PS**0.52125*TS**(-0.60438)      ABS02150
EH(14,I)=0.772E-04*PPMNO2*WA(I)*PS**0.18066*TS**0.20911      ABS02160
C***K=15 FOR ASL 3.3 - 4.3_ MICRON H2O CONTINUUM      ABS02170
EH(15,I)=PPW*0      ABS02180
C***K=16 FOR 4.6 - 4.8 MICRON H2O CONTINUUM      ABS02190
EH(16,I)=PPW*(PPW+3.0E-03*(PS-PPW))      ABS02200
IF (I.EQ.NL) GO TO 2300      ABS02210
IF (MODEL.EQ.0.AND.I.GE.1) GO TO 3600      ABS02220
T2=T(I+1)
W2=WH(I+1)
PPW=4.56E-6*W2*T2      ABS02230
2300 IF (H1.GE.Z(I)) J1=I      ABS02240
IF ((JP.EQ.0).AND.(.NOT.ISPOT))      ABS02250
1WRITE (1000,8300)I,Z(I),(EH(K,I),K=1,6),EH(8,I),(EH(K,I),K=10,14)      ABS02260
2400 CONTINUE      ABS02270
X1=H1      ABS02280
CALL POINT (H1,N,NP1,TX)      ABS02290
J1=N      ABS02300
DO 2500 K=1,KMAX      ABS02310
EK(K)=TX(K)      ABS02320
JEXTRA=0      ABS02330
JMIN=0      ABS02340
C**ITYPE=1 MEANS HORIZONTAL PATH ****      ABS02350
IF (ITYPE.EQ.1) GO TO 3600      ABS02360
IF (ITYPE.EQ.3) H2=Z(NL)      ABS02370
C** ANGLE GREATER THAN 90 DEGREES MEANS DOWNWARD TRAJECTORY ****      ABS02380
IF (ANGLE.GT.90.0) GO TO 3800      ABS02390
C** IF THE PATH IS NOT HORIZONTAL OR DOWNWARD THEN IT IS UPWARD TRAJECTORY      ABS02400
2600 IF (ANGLE.GT.90.0.AND.NP1.GT.0) J1=J1+1      ABS02410
J2=NL      ABS02420
IF (ITYPE.EQ.3) GO TO 2700      ABS02430
CALL POINT (H2,N,NP,TX)      ABS02440
J2=N      ABS02450
IF (NP.GT.0) J2=J2-1      ABS02460
2700 DO 2800 K=1,KMAX      ABS02470
IF (K.EQ.9.OR.K.EQ.7) GO TO 2800      ABS02480
EH(K,J1)=E(K)      ABS02490
IF (ITYPE.EQ.3) GO TO 2800      ABS02500
EH(K,J2+1)=TX(K)      ABS02510
2800 CONTINUE      ABS02520
C**** NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)      ABS02530
IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (1000,8400)      ABS02540
DO 2900 K=1,KMAX      ABS02550
WK(K)=0      ABS02560
DO 3500 I=J1,J2      ABS02570
X1=Z(I)      ABS02580
IF (I.LT.NL) X2=Z(I+1)      ABS02590
IF (I.EQ.NL) X2=Z(I)      ABS02600
IF (I.EQ.J1) X1=H1      ABS02610
IF (I.EQ.J2) X2=H2      ABS02620
DZ=X2-X1      ABS02630
IF (I.EQ.NL) DZ=Z(I)-Z(I-1)      ABS02640
DS=DZ      ABS02650
C**** UPWARD TRAJECTORY      ABS02660
RX=(RE+X1)/(RE+X2)      ABS02670
THETA= ASIN(SPHI)/CA      ABS02680
PHI= ASIN(SPHI*RX)/CA      ABS02690
BET=THETA-PHI      ABS02700
SALP=RX*SPHI      ABS02710
IF (SPHI.GT.1.E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI      ABS02720
BETA=BET+BET      ABS02730
PSI=BETA+PHI-ANGLE      ABS02740
PHI=180.-PHI      ABS02750
SR=SR+DS      ABS02760
JEXTRA=0      ABS02770
DO 3400 K=1,KMAX      ABS02780
ABS02790
ABS02800

```

```

IF (K.EQ.7.OR.K.EQ.9) GO TO 3100          ABS02810
EV=DS*EH(K,I)                         ABS02820
IF (I.EQ.NL) GO TO 3000          ABS02830
IF (EH(K,I).EQ.0.0.DR.EH(K,I+1).EQ.0.0) GO TO 3100  ABS02840
IF (EH(K,I).EQ.EH(K,I+1)) GO TO 3200          ABS02850
EV=DS*(EH(K,I)-EH(K,I+1))/ALOG(EH(K,I)/EH(K,I+1))  ABS02860
GO TO 3200          ABS02870
3000 IF (EH(K,I).EQ.0.0) GO TO 3100          ABS02880
IF (EH(K,I-1).EQ.0.0) GO TO 3100          ABS02890
IF (EH(K,I).EQ.EH(K,I-1)) GO TO 3200          ABS02900
EV=EV/ALOG(EH(K,I-1)/EH(K,I))  ABS02910
GO TO 3200          ABS02920
3100 EV=0          ABS02930
3200 VH(K)=VH(K)+EV          ABS02940
IF (I.EQ.JSTAR) GO TO 3300          ABS02950
WLAY(I,K)=EV+W(K)          ABS02960
W(K)=0          ABS02970
GO TO 3400          ABS02980
3300 W(K)=EV          ABS02990
IF (J1.NE.J2) GO TO 3400          ABS03000
WLAY(J2+1,K)=W(K)          ABS03010
W(K)=0          ABS03020
JEXTRA=1          ABS03030
3400 CONTINUE          ABS03040
1 IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8500) I,
     X1,(VH(L),L=1,6),VH(8),PSI,PHI,BETA,THETA,SR  ABS03050
1 IF (I.GE.NL) GO TO 3500          ABS03060
SPHI=SPhi*RX          ABS03070
IF (SALP.GE.1.) SPhi=SALP          ABS03080
3500 CONTINUE          ABS03090
GO TO 5800          ABS03100
C**** HORIZONTAL PATH          ABS03110
3600 DO 3700 K=1,KMAX          ABS03120
IF (K.EQ.7.OR.K.EQ.9) GO TO 3700          ABS03130
W(K)=RANGE*EH(K,1)          ABS03140
IF (MODEL.GT.0) W(K)=RANGE*TX(K)          ABS03150
VH(K)=W(K)          ABS03160
3700 CONTINUE          ABS03170
GO TO 6100          ABS03180
3800 CONTINUE          ABS03190
C**** DOWNWARD TRAJECTORY          ABS03200
K2=0          ABS03210
IF (NP1.EQ.1) J1=J1-1          ABS03220
IF (J1.LE.0) J1=1          ABS03230
J2=J1+1          ABS03240
J=J1+1          ABS03250
IF (H2.GT.Z(J1+1).OR.H1.EQ.H2) GO TO 4000  ABS03260
IF (NP1.EQ.1.AND.H2.GE.Z(J1+1)) GO TO 4000  ABS03270
CALL POINT (H2,N,NP2,TX)          ABS03280
DO 3900 K=1,KMAX          ABS03290
3900 W(K)=TX(K)          ABS03300
IF (H2.LT.H1) H=H2          ABS03310
J2=N          ABS03320
4000 A0=(RE+H1)*SPhi          ABS03330
DO 4100 I=1,J1          ABS03340
HMIN=A0-RE          ABS03350
JMIN=I          ABS03360
IF (HMIN.LE.Z(I+1)) GO TO 4200  ABS03370
4100 CONTINUE          ABS03380
4200 X=HMIN          ABS03390
IF (HMIN.LE.0) GO TO 4400          ABS03400
CALL POINT (X,N,NP,TX)          ABS03410
JMIN=N          ABS03420
HMIN=A0-RE          ABS03430
IF (ABS(X-HMIN).GT.0.0001) GO TO 4200  ABS03440
IF (H2.GE.H1) J2=N          ABS03450
IF (H2.GE.H1.OR.H2.LT.HMIN) H=HMIN          ABS03460
IF (.NOT.ISPOT) WRITE (IOOUT,8600) HMIN  ABS03470
IF (H2.LT.HMIN) J2=N          ABS03480
IF ((H2.LT.HMIN).AND(.NOT.ISPOT)) WRITE (IOOUT,8700) HMIN  ABS03490
IF ((H2.LT.HMIN).AND(.NOT.ISPOT)) WRITE (IOOUT,8700) HMIN  ABS03500

```

```

4400 GO TO 4500                                ABS03510
IF (.NOT.ISPOT) WRITE (I00UT,8600) HMIN        ABS03520
IF (<H2,LT,H1>) GO TO 4500                  ABS03530
IF <(ITYPE.EQ.3.OR.H2.GE.H1).AND.(&.NOT.ISPOT)>
,      WRITE (I00UT,8800)                          ABS03540
,      ITYPE=2                                    ABS03550
JMIN=0                                         ABS03560
J2=1                                           ABS03570
H2=0.0                                         ABS03580
H=0.0                                         ABS03590
ABS03600
C**** NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)
4500 IF <(JP.EQ.0).AND.(&.NOT.ISPOT)> WRITE (I00UT,8400)
JSTOR=J-1                                     ABS03610
DO 5100 I=1,NL                                 ABS03620
J=J-1                                         ABS03630
IF (I.NE.1) X1=Z(J+1)                         ABS03640
X2=Z(J)                                       ABS03650
IF (<J.EQ.J2.AND.K2.EQ.0>) X2=H              ABS03660
IF (<J.EQ.JMIN.AND.K2.EQ.1>) X2=HMIN        ABS03670
HM=(RE+X1)*SPHI-RE                           ABS03680
IF (<HM.GT.Z(J).AND.HM.GT.X2>) X2=HM        ABS03690
RX=(RE+X1)/(RE+X2)                           ABS03700
DS=X1-X2                                      ABS03710
ALP=90.0                                       ABS03720
DS=RE-X2                                      ABS03730
ALP=90.0                                       ABS03740
THET= ASIN(SPHI)/CA                          ABS03750
SALP=RX*SPHI                                  ABS03760
IF (<ABS(Z-HM).GT.1.0E-5>) ALP= ASIN(SALP)/CA
BET=ALP-THET                                  ABS03770
IF (<SPHI.GT.1.0E-10>) DS=(RE+X2)*SINK(BET*CA)/SPHI
THETA=180.0-THET                            ABS03780
ABS03790
BETA=BETA+BET                                 ABS03800
PSI=BETA-ALP-ANGLE+180.0                      ABS03810
SR=SR+DS                                     ABS03820
ABS03830
DO 5000 K=1,KMAX                            ABS03840
IF (<K.EQ.7.OR.K.EQ.9>) GO TO 5000          ABS03850
AJ=EH(K,J)                                   ABS03860
BJ=EH(K,J+1)                                   ABS03870
IF (<J.EQ.J1>) BJ=EK(K)                      ABS03880
IF (<J.EQ.J2.AND.H2.LT.H1.AND.H2.GT.0.0>) AJ=WK(K)
IF (<J.EQ.JMIN.AND.H2.GE.H1>) AJ=TX(K)       ABS03890
IF (<J.EQ.JMIN.AND.ABS(H2-HM).LT.1.0E-5>) AJ=TX(K)
IF (<K2.EQ.0>) GO TO 4600                   ABS03900
IF (<J.EQ.J2>) BJ=W(K)                       ABS03910
IF (<J.EQ.JMIN>) AJ=TX(K)                   ABS03920
4600 IF (<AJ.EQ.0.0.OR.BJ.EQ.0.0>) GO TO 4800
IF (<AJ.EQ.BJ>) GO TO 4700                 ABS03930
EV=DS*(AJ-BJ)/ALOG(AJ/BJ)                   ABS03940
GO TO 4900                                     ABS03950
4700 EV=DS*AJ                                 ABS03960
GO TO 4900                                     ABS03970
4800 EV=0.0                                     ABS03980
4900 VH(K)=VH(K)+EV                          ABS03990
5000 WLAYC(J,K)=EV                          ABS04000
,      IF <(JP.EQ.0).AND.(&.NOT.ISPOT)> WRITE (I00UT,8500) J,
,      X1,(VH(L),L=1,6),VH(8),PSI,ALP,BETA,THETA,SR
,      IF (<J.EQ.J2.AND.H2.GE.H1>) GO TO 5600
,      IF (<J.EQ.JMIN.AND.K2.EQ.1>) GO TO 5400
,      SPHI=SALP
,      IF (<J.EQ.J2.AND.K2.EQ.0>) GO TO 5200
5100 CONTINUE                                 ABS04010
5200 IF (<HMIN.LE.0>) GO TO 5800             ABS04020
IF <(LEN.EQ.0).AND.(&.NOT.ISPOT)> WRITE (I00UT,8900)
IF <(LEN.EQ.1).AND.(&.NOT.ISPOT)> WRITE (I00UT,9000)
IF (<LEN.EQ.0>) GO TO 5800                 ABS04030
K2=1                                         ABS04040
X1=X2                                         ABS04050
IF (<ABS(X1-HMIN).LE.0.001>) GO TO 5800
H=HMIN                                       ABS04060
J=J2+1                                         ABS04070
IF (<NP2.EQ.1>) J=J-1                        ABS04080
ABS04090
ABS04100
ABS04110
ABS04120
ABS04130
ABS04140
ABS04150
ABS04160
ABS04170
ABS04180
ABS04190
ABS04200

```

```

B=BETA
PH=180.0- ASIN(SPHI)/CA
TS=SR
PS=PSI
DO 5300 K=1,KMAX
5300 E(K)=VH(K)
GO TO 4500
5400 BETA=2.*BETA-B
PSI=2.*PSI-PS
SR=2.*SR-TS
C   LONG PATH TAKEN
    PHI=PH
    DO 5500 K=1,KMAX
5500 VH(K)=2.*VH(K)-E(K)
    GO TO 5800
5600 DO 5700 K=1,KMAX
5700 VH(K)=2.0*VH(K)
    BETA=2.0*BETA
    SR=2.0*SR
    IF (H2.EQ.H1) GO TO 5800
    SPHI=SIN(ANGLE*CA)
    GO TO 2600
5800 CONTINUE
    DO 5900 K=1,KMAX
5900 W(K)=VH(K)
5900 CONTINUE
6000 CONTINUE
6100 RETURN
C
6200 FORMAT (7F10.3)
6250 FORMAT (2F10.3,12)
6300 FORMAT (<1H0,9X,4H H1=,F7.3,6H KM,H2=,F7.3,9H KM,ANGLE=,
1      F8.4,13H GEOM. RANGE =,F7.2,8H KM,BETA=,F8.5,
2      5H,YIS=,F6.1>)
6400 FORMAT (3F10.3,2F5.1,2E10.3,2F10.3)
6500 FORMAT (<10X,26H INPUT METEOROLOGICAL DATA:/10X,2H Z=,
1      F7.2,7H KM, P=,F7.2,6H MB,T=,F5.1,8H C, DEW ,
2      7H PT TEMP,F5.1,17H C, REL HUMIDITY=,F5.1,
3      16H %, H20 DENSITY=,1PE9.2,7H GM M-3/10X,
4      15H OZONE DENSITY=,E9.2,20H GM-3, VISUAL RANGE=
5      ,0PF6.1,10H KM,RANGE=,F10.3,4H KM >)
6600 FORMAT (<24H MODEL ATMOSPHERE NO. 7./4X,6H Z (KM),3X,
1      6H P (MB),4X,30H (C) DEW PT %RH H20(GM.M-3) ,
2      19H 03(CM.M-3) NO. DEN. >)
6700 FORMAT (</10X,28H HORIZONTAL PATH, ALTITUDE =,F7.3,
1      11H KM,RANGE =,F7.3,3H KM>)
6800 FORMAT (</10X,37H SLANT PATH BETWEEN ALTITUDES H1 AND ,
1      13HH2 WHERE H1 =,F7.3,8H KM H2 =,F7.3,
2      18H KM,ZENITH ANGLE =,F7.3,8H DEGREES>)
6900 FORMAT (</10X,39H SLANT PATH TO SPACE FROM ALTITUDE H1 =
1      ,F7.3,19H KM, ZENITH ANGLE =,F7.3,
2      8H DEGREES>)
7200 FORMAT (</20X,18H MODEL ATMOSPHERE ,I1,11H = TROPICAL>)
7300 FORMAT (</20X,18H MODEL ATMOSPHERE ,I1,
1      21H = MIDLATITUDE SUMMER>)
7400 FORMAT (</20X,18H MODEL ATMOSPHERE ,I1,
1      21H = MIDLATITUDE WINTER>)
7500 FORMAT (</20X,18H MODEL ATMOSPHERE ,I1,14H = SUB-ARCTIC ,
1      7H SUMMER >)
7600 FORMAT (</20X,18H MODEL ATMOSPHERE ,I1,14H = SUB-ARCTIC ,
1      7H WINTER >)
7700 FORMAT (</20X,18H MODEL ATMOSPHERE ,I1,11H = 1962 US ,
1      10H STANDARD >)
7800 FORMAT (</20X,18H MODEL ATMOSPHERE ,I1,20H = ISRAELI STANDARD ,
1      16H(YEAR, DAYTIME)>)
7900 FORMAT (</20X,18H MODEL ATMOSPHERE ,I1,20H = ISRAELI STANDARD ,
1      18H(YEAR, NIGHTTIME)>)
8000 FORMAT (</10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO ,
1      4H Y2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1 <
2      ,F6.2,3H - ,F5.2,10H MICRONS >>)

```

8100 FORMAT (1H1,///10X,20H HORIZONTAL PROFILES/)	ABS04910
8200 FORMAT (38H TRAJECTORY MISSES EARTHS ATMOSPHERE,	ABS04920
1 31HCLOSEST DISTANCE OF APPROACH IS,F10.2,1X,/	ABS04930
2 1X,18HEND OF CALCULATION)	ABS04940
8300 FORMAT (1X,14,F6.1,12(E9.3))	ABS04950
8400 FORMAT (1H1,///10X,21H VERTICAL PROFILES ,53X,3HPSI,	ABS04960
1 6X,3HPHI,6X,4HBETA,3X,13HTHETA RANGE)	ABS04970
8500 FORMAT (13,F6.1,7E10.3,4F9.4,F6.1)	ABS04980
8600 FORMAT (8H HMIN = ,F10.3)	ABS04990
8700 FORMAT (40H H2 WAS SET LESS THAN HMIN AND HAS BEEN	ABS05000
1 34HRESET EQUAL TO HMIN I.E. H2 = ,F10.3)	ABS05010
8800 FORMAT (41H PATH INTERSECTS EARTH - PATH CHANGED TO	ABS05020
1 23HTYPE 2 WITH H2 = 0.0 KM)	ABS05030
8900 FORMAT (36H CHOICE OF TWO PATHS FOR THIS CASE -	ABS05040
1 42HSHORTEST PATH TAKEN. FOR LONGER PATH SET	ABS05050
2 6HLEN=1,)	ABS05060
9000 FORMAT (44H CHOICE OF TWO PATHS FOR THIS CASE -LONGEST	ABS05070
1 41HPATH TAKEN. FOR SHORT PATH SET LEN = 0)	ABS05080
END	ABS05090

```

SUBROUTINE CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN)      CKR00010
LOGICAL ISPOT                                         CKR00020
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU   CKR00030
IERR=0                                                 CKR00040
IF (<MULDV.LE.0>) MULDV=1                                CKR00050
C     CHECK FOR PROPER WINDOW REGION                      CKR00060
IF (<V2.LE.830.0>) GO TO 1090                           CKR00070
IF (<V1.GE.1250.0.AND.V2.LE.2010.0>) GO TO 1090           CKR00080
IF (<V1.GE.3330.0.AND.V2.LT.5010.0>) GO TO 1090           CKR00090
IF (<V1.GE.39990.0>) GO TO 1090                         CKR00100
GO TO 1100                                              CKR00110
1090 WRITE<(I0OUT,3190)
TRAN=1.
IERR=1
RETURN
C     CHECK FOR PROPER INTEGER VALUES                     CKR00150
1100 IV1=20*IFIX(<V1-10.0>/20.0)+10                  CKR00160
IV2=20*IFIX(<V2-10.0>/20.0+0.99)+10                CKR00170
IF (<IV1.LT.830>) IV1=830                            CKR00180
IF (<IV1.LT.1250.AND.IV2.GT.1250>) IV2=1250          CKR00190
IF (<IV1.LT.2010.AND.IV2.GT.2010>) IV1=2010          CKR00210
IF (<IV1.LT.3330.AND.IV2.GT.3330>) IV2=3330          CKR00220
IF (<IV1.LT.5010.AND.IV2.GT.5010>) IV1=5010          CKR00230
IF (<IV1.LT.39990.AND.IV2.GT.39990>) IV2=39990        CKR00240
V1=FLOAT(IV1)
V2=FLOAT(IV2)
DV=20.*FLOAT(MULDV)
IWVCK=0
IDV=IFIX(DV)
WHEN CALLED FROM SPOT CHECK FOR MORE THAN 15 DIVISIONS      CKR00290
C     (16 WAVENUMBER VALUES) WHICH IS ARRAY SIZE.            CKR00300
C     CHECK HERE FOR ROUNDING PROBLEMS ON V1, V2 CAUSING      CKR00310
C     TOO SMALL AN INCREMENT                                 CKR00320
C     IF (<ISPOT.AND.<FLOAT(IV2-IV1)/15..GT.<FLOAT(IDV)+.001>>>) IWVCK=1 CKR00330
C     IF (<IWVCK.NE.1>) GO TO 91                           CKR00340
C     CKDV=FLOAT(IV2-IV1)/<15.*20.>
MULDV=IFIX(CKDV)
IF (<FLOAT(MULDV)/CKDV.LT..99>) MULDV=MULDV+1          CKR00350
DVHOLD=20.*FLOAT(MULDV)
WRITE<(I0OUT,93> DV,DVHOLD
DV=DVHOLD
IF (<DV.LT.20.>) DV=20.
IDV=IFIX(DV)
93  FORMAT<1X,'DIVISION LIMITS CHANGED FROM ',F10.3,          CKR00440
+ ' TO ',F10.3>
91  CONTINUE
RETURN
3190 FORMAT<6X,'*****FREQUENCY IS OUTSIDE OF THE WINDOW*****'/> CKR00450
1       6X,'*****TOTAL TRANSMITTANCE IS 1.0000*****'>          CKR00460
END                                              CKR00470
                                                CKR00480
                                                CKR00490
                                                CKR00500

```

```

SUBROUTINE FREQSL(I, IV, W, TX)          FRE00010
COMMON /M05/ C1(501),C2(258),C3(86),C4(33),C5(6),C5DUM(9),C8(82),   FRE00020
      C11(4),C12(15),C14(21),C15(6)      FRE00030
COMMON /M07/ TR(67),FW(67),FO(67)        FRE00040
COMMON /M08/SUM4,SUM5,SUM8,SUM11,SUM6    FRE00050
DIMENSION W(16),TX(16)                   FRE00060
1 IF (<I.EQ.1>) GO TO 10                FRE00070
IF (<I.GE.2.AND.I.LE.3>) GO TO 11        FRE00080
IF (<I.GE.4.AND.I.LE.5>) GO TO 12        FRE00090
IF (<I.GE.6.AND.I.LE.12>) GO TO 13        FRE00100
IF (<I.GE.13.AND.I.LE.21>) GO TO 15        FRE00110
IF (<I.EQ.22>) GO TO 16                FRE00120
IF (<<I.GE.23.AND.I.LE.59>.OR.<I.GE.127.AND.I.LE.209>>) RETURN  FRE00130
IF (<<I.GE.60.AND.I.LE.63>>) GO TO 14        FRE00140
IF (<I.GE.64.AND.I.LE.76>) GO TO 18        FRE00150
IF (<<I.GE.77.AND.I.LE.81>.OR.<I.GE.87.AND.I.LE.96>>) GO TO 18  FRE00160
IF (<I.GE.82.AND.I.LE.86>) GO TO 30        FRE00170
IF (<<I.GE.97.AND.I.LE.101>.OR.<I.GE.105.AND.I.LE.109>>) GO TO 14  FRE00180
IF (<I.GE.102.AND.I.LE.104>) GO TO 9        FRE00190
IF (<I.GE.110.AND.I.LE.112>) GO TO 21        FRE00200
IF (<I.GE.113.AND.I.LE.123>) GO TO 22        FRE00210
IF (<<I.GE.124.AND.I.LE.126>.OR.<I.GE.210.AND.I.LE.363>>) GO TO 23  FRE00220
IF (<<I.GE.364.AND.I.LE.419>.OR.<I.GE.454.AND.I.LE.599>>) GO TO 24  FRE00230
IF (<<I.GE.420.AND.I.LE.453>.OR.<I.GE.600.AND.I.LE.606>.OR.<I.GE.  FRE00240
1.1160.AND.I.LE.1334>>) GO TO 35
IF (<I.GE.607.AND.I.LE.609>) GO TO 25        FRE00250
IF (<I.GE.610.AND.I.LE.621>) GO TO 26        FRE00260
IF (<<I.GE.622.AND.I.LE.629>.OR.<I.GE.686.AND.I.LE.1159>.OR.<I.GE.  FRE00270
1.1335>>) GO TO 27
1 IF (<I.GE.630.AND.I.LE.685>) GO TO 28        FRE00280
4 CALL H20VAP(I,W,C1,TX)                  FRE00290
GO TO 40
6 CALL OZONE(I,W,C3,TX)                  FRE00300
7 CALL UNIMIX(I,W,C2,TX)                  FRE00310
GO TO 4
8 CALL H02(I,W,C15,TX)                  FRE00320
GO TO 6
9 CALL H20410(I,IV,W,C5,TX,SUM5)        FRE00330
GO TO 8
10 CALL NH3(I,W,C14,TX)                  FRE00340
GO TO 9
11 CALL NITRICK(I,W,C11,SUM11,TX)        FRE00350
GO TO 10
12 CALL NITRICK(I,W,C11,SUM11,TX)        FRE00360
13 CALL NH3(I,W,C14,TX)                  FRE00370
14 CALL H20410(I,IV,W,C5,TX,SUM5)        FRE00380
GO TO 6
15 CALL S02(I,W,C12,TX)                  FRE00390
GO TO 13
16 CALL S02(I,W,C12,TX)                  FRE00400
GO TO 14
18 CALL H20410(I,IV,W,C5,TX,SUM5)        FRE00410
19 CALL NITRO(I,W,C4,TX,SUM4)            FRE00420
GO TO 6
20 CALL S02(I,W,C12,TX)                  FRE00430
GO TO 12
21 CALL MOLSC(T,IV,W,TX,SUM6)           FRE00440
GO TO 14
22 CALL MOLSC(T,IV,W,TX,SUM6)           FRE00450
GO TO 6
23 CALL MOLSC(T,IV,W,TX,SUM6)           FRE00460
GO TO 7
24 CALL MOLSC(T,IV,W,TX,SUM6)           FRE00470
GO TO 4
25 CALL UNIMIX(I,W,C2,TX)                FRE00480
35 CALL MOLSC(T,IV,W,TX,SUM6)           FRE00490
GO TO 40
26 CALL UV0ZNE(I,W,C8,TX,SUM8)          FRE00500
GO TO 25
27 CALL UV0ZNE(I,W,C8,TX,SUM8)          FRE00510
FRE00520
FRE00530
FRE00540
FRE00550
FRE00560
FRE00570
FRE00580
FRE00590
FRE00600
FRE00610
FRE00620
FRE00630
FRE00640
FRE00650
FRE00660
FRE00670
FRE00680
FRE00690
FRE00700

```

```
CALL MOLSCT(IV,W,TX,SUM6)          FRE00710
28  GO TO 40                      FRE00720
CALL H2OVAP(I,W,C1,TX)             FRE00730
GO TO 27                      FRE00740
30  CALL H2O410(I,IV,W,C5,TX,SUM5) FRE00750
GO TO 20                      FRE00760
40  RETURN                         FRE00770
END                            FRE00780
```

```

SUBROUTINE H2OVAP(I,W,C1,TX) H2V00010
DIMENSION C1(501),TX(1),W(1) H2V00020
C*****TRANSMITTANCE FOR WATER VAPOR***** H2V00030
IF (W(1).LT.1.0E-20) GO TO 500 H2V00040
IF (I.LE.22) II=I H2V00050
IF (I.GE.60.AND.I.LE.126) II=I-37 H2V00060
IF(I.GE.210.AND.I.LE.419) II=I-120 H2V00070
IF(I.GE.454.AND.I.LE.599) II=I-154 H2V00080
IF (I.GE.630) II=I-184 H2V00090
WS1=ALOG10(W(1))+C1(II) H2V00100
TX(1)=EXP(-10*(-1.14619+0.55013*WS1)) H2V00110
500 RETURN H2V00120
END H2V00130

```

```

SUBROUTINE H20410(I,IV,W,C5,TX,SUM5) H2F00010
C***WATER VAPOR CONTINUUM, 3-5 AND 8-12 MICRON REGIONS H2F00020
DIMENSION C5(6),TX(5),W(16) H2F00030
V=FLOAT(IV) H2F00040
IF(I.GT.109) GO TO 100 H2F00050
IF(I.GT.22) GO TO 200 H2F00060
C***1.LE.I.LE.22:CALCULATE OPTICAL DEPTH DUE TO 8-12 MICRON CONTINUUM H2F00070
TX(5)=(4.18+5578.0*EXP(-7.87E-03*V))*W(5) H2F00080
GO TO 300 H2F00090
200 CONTINUE H2F00100
IF(I.LT.63) GO TO 100 H2F00110
IF(I.GT.68) GO TO 400 H2F00120
C***63.LE.I.LE.68:CALCULATE OPTICAL DEPTH DUE TO 4.6-4.8 MICRON H2F00130
C***CONTINUUM MODEL FROM BEN-SHALOM ET AL.1980,SPIE,VOL.253,261. H2F00140
II=I-62 H2F00150
TX(5)=C5(II)*577.6*W(16) H2F00160
GO TO 300 H2F00170
400 CONTINUE H2F00180
IF(I.LT.77) GO TO 100 H2F00190
C***77.LE.I.LE.109:CALCULATE OPTICAL DEPTH DUE TO 3.3-4.3 MICRON H2F00200
C***CONTINUUM MODEL FROM WATKINS ET AL.1979,APPL.OPT.,VOL.18,1149. H2F00210
V=V*1.0E-03 H2F00220
V2=V*V H2F00230
V3=V2*V H2F00240
CBURCH=46.4745-48.0898*V+16.3988*V2-1.83217*V3 H2F00250
CASL=-370.082+508.137*V-225.822*V2+32.7744*V3 H2F00260
TX(5)=CBURCH*W(10)+CASL*W(15) H2F00270
GO TO 300 H2F00280
100 TX(5)=0.0 H2F00290
300 SUM5=TX(5) H2F00300
IF(TX(5).LT.1.0E-05) GO TO 500 H2F00310
IF(TX(5).GT.20.0) GO TO 600 H2F00320
TX(5)=EXP(-TX(5)) H2F00330
RETURN H2F00340
500 TX(5)=1.0 H2F00350
RETURN H2F00360
600 TX(5)=0.0 H2F00370
RETURN H2F00380
END H2F00390

```

```

SUBROUTINE LTPATH(WLAY,WPATH,TBBy,ANGLE,LEN,ITYPE,H1,H2,MODEL)      LTP00010
COMMON /M01/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL      LTP00020
DIMENSION WLAY(34,16),TBBy(68),WPATH(68,16)                         LTP00030
COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1                      LTP00040
COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL                          LTP00050
COMMON /I0UNIT/I0IN, I0OUT,IPHFLN,L0UNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUL LTP00060
COMMON /SPOTLO/ISPOT,LOREAD,N16                                         LTP00070
LOGICAL ISPOT,N16,LOREAD                                           LTP00080
555 IL=0                                                       LTP00090
IF (ITYPE.EQ.1) GO TO 1000                                         LTP00100
IF (ITYPE.EQ.2.AND.H1.EQ.H2) J2=J1                                LTP00110
IF (H2.GT.H1.AND.ANGLE.GT.90.0.AND.NP1.EQ.1) J1=J1-1              LTP00120
IF (JEXTRA.EQ.1) J2=J2+1                                         LTP00130
IF ((ITYPE.EQ.2).AND.(H1.GT.H2).AND.(LENTOR.EQ.1)) J2=          LTP00140
   J2-1
IF (ITYPE.EQ.3) J2=NLL                                         LTP00150
IF (.NOT.ISPOT) WRITE (I00UT,1200)                                     LTP00160
DO 100 IK=1,68                                         LTP00170
TBBy(IK)=0.                                                 LTP00180
DO 100 K=1,KMAX                                         LTP00190
WPATH(IK,K)=0.                                              LTP00200
100 CONTINUE                                         LTP00210
LEN=0                                                       LTP00220
NLL=NL-1                                         LTP00230
IL=J1+1                                         LTP00240
IJ=IL+NLL                                         LTP00250
DO 200 K=1,KMAX                                         LTP00260
E(K)=0.                                                 LTP00270
200 CONTINUE                                         LTP00280
IF (ANGLE.GT.90.0) GO TO 300                                         LTP00290
LEN=1.                                                 LTP00300
IL=J1-1                                         LTP00310
HMIN=1.0E-6                                         LTP00320
IJ=NLL                                         LTP00330
300 CONTINUE                                         LTP00340
DO 800 IK=1,68                                         LTP00350
IF (LEN.EQ.0) IL=IL-1                                         LTP00360
IF (LEN.EQ.1) IL=IL+1                                         LTP00370
IJ=IJ-1                                         LTP00380
IF (IL.EQ.0) GO TO 800                                         LTP00390
DO 400 K=1,KMAX                                         LTP00400
W(K)=E(K)+WLAY(IL,K)                                         LTP00410
WPATH(IK,K)=W(K)                                         LTP00420
400 CONTINUE                                         LTP00430
IF (IL.LE.0.OR.IL.GE.NL) GO TO 500                           LTP00440
TBAR=(T(IL)+T(IL+1))*0.5                                     LTP00450
C JEXTRA IS 1 ONLY WHEN PROGRAM NEVER LEAVES ONE LAYER           LTP00460
IF (JEXTRA.EQ.1) TBAR=(T(J1)+T(J1+1))*0.5                  LTP00470
500 CONTINUE                                         LTP00480
TBBy(IK)=TBAR                                         LTP00490
DO 600 K=1,KMAX                                         LTP00500
E(K)=W(K)                                         LTP00510
600 CONTINUE                                         LTP00520
IF (ANGLE.LE.90.0.AND.IL.EQ.NLL) GO TO 900               LTP00530
IF (ITYPE.EQ.3.AND.ANGLE.LE.90.0) GO TO 700               LTP00540
IF (ITYPE.EQ.3.AND.LEN.EQ.1.AND.IL.EQ.J2) GO TO 900       LTP00550
IF (ITYPE.EQ.2.AND.LENTOR.EQ.0.AND.IL.EQ.J2) GO TO 900       LTP00560
IF (IL.EQ.JMIN.AND.HMIN.GT.0) LEN=1                      LTP00570
IF (IL.EQ.1.AND.HMIN.LE.0.0) GO TO 900                   LTP00580
IF (LEN.EQ.0) GO TO 700                                         LTP00590
IF (IL.EQ.JMIN.AND.IJ.EQ.IL+NLL) IL=IL-1                 LTP00600
IF (ITYPE.EQ.2.AND.IL.EQ.J2) GO TO 900                   LTP00610
700 CONTINUE                                         LTP00620
IF (.NOT.ISPOT) WRITE (I00UT,1300) IK,(WPATH(IK,K),K=          LTP00630
   1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBy(IK)           LTP00640
800 CONTINUE                                         LTP00650
IKMAX=68                                         LTP00660
LEN=LENTOR                                         LTP00670
RETURN                                         LTP00680
900 CONTINUE                                         LTP00690

```

```

1 IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=
1 ,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK) LTP00710
IKMAX=IK LTP00720
LEN=LENTOR LTP00730
RETURN LTP00740
1000 DO 1100 K=1,KMAX LTP00750
WPATH(1,K)=W(K) LTP00760
1100 CONTINUE LTP00770
IF (MODEL.EQ.0) JI=1 LTP00780
J2=JI LTP00790
TBBY(1)=T(J1) LTP00800
IKMAX=1 LTP00810
IK=1 LTP00820
IF (.NOT.ISPOT) WRITE (IOOUT,1200) LTP00830
1 IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=
1 ,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK) LTP00840
HMIN=1.0E-6 LTP00850
RETURN LTP00860
LTP00870
LTP00880
LTP00890
C 1200 FORMAT (//,20X,37H CUMULATIVE ABSORBER AMOUNTS FOR THE , LTP00900
1 16HATMOSPHERIC PATH,/,8X,3HH20,5X,4HC02+,6X, LTP00910
2 2H03,7X,2HN2,6X,5HH20 C,4X,5HMOL S,4X,5H03 UV, LTP00920
3 4X,5HH20 C,5X,4HHN03,6X,3HS02,6X,3HNN3,6X,3HN02, LTP00930
4 5X,4HTAVE) LTP00940
1300 FORMAT (I5,12E9.3,F10.3) LTP00950
END LTP00960

```

```

SUBROUTINE MOLSCT(IV,W,TX,SUM6)                               MOL00010
C*****TRANSMITTANCE FOR MOLECULAR SCATTERING *****          MOL00020
C   C6 EXPRESSION MODIFIED AS PER SHETTLE ET AL 1980,APPL.OPT.,VOL.19,
C   2875.                                                 MOL00030
DIMENSION TX(6),W(6)                                         MOL00040
V=FLOAT(IV)                                              MOL00050
C6=(V**4)/(9.67578E+18-1.11836E+09*V**2)                 MOL00060
TX(6)=C6*W(6)                                              MOL00070
SUM6=TX(6)                                              MOL00080
IF (TX(6).EQ.0.0) GO TO 200                                MOL00090
IF (TX(6).LE.0.1) GO TO 100                                MOL00110
IF (TX(6).GT.20.) GO TO 300                                MOL00120
TX(6)=EXP(-TX(6))                                         MOL00130
GO TO 400                                              MOL00140
100 TX(6)=1.0-TX(6)+0.5*TX(6)*TX(6)                      MOL00150
GO TO 400                                              MOL00160
200 TX(6)=1.0                                         MOL00170
GO TO 400                                              MOL00180
300 TX(6)=0.0                                         MOL00190
400 RETURN                                              MOL00200
END                                                       MOL00210

```

```

SUBROUTINE NH3(I,W,C13,TX) NHA00010
COMMON /M03/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FN02(9), NHA00020
1 O1(9),O2(9),PPMS02,PPMNH3,PPMN02 NHA00030
DIMENSION C13(21),TX(13),W(13) NHA00040
***** NHA00050
C THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NH3 ( PPM READ IN NHA00060
THE MAIN PROGRAM). NHA00070
***** NHA00080
***** NHA00090
***** NHA00100
1 I1=I NHA00110
IF (W(13).LT.1.0E-20) GO TO 3 NHA00120
WS13=ALOG10(W(13))+C13(I1) NHA00130
DO 1 J=1,9 NHA00140
IF (WS13-FNH3(J)).GT.2.2,1 NHA00150
CONTINUE NHA00160
TX(13)=EXP(-10**((FH1(J)+FH2(J))*WS13)) NHA00170
RETURN NHA00180
END NHA00190

```

```

SUBROUTINE NITRIC(I,W,C11,SUM11,TX)          NITC0010
DIMENSION C11(4),TX(11),W(11)                 NITC0020
C***** TRANSMITTANCE FOR NITRIC ACID*****NITC0030
HABS=0.                                         NITC0040
IF (I.LT.2.OR.I.GT.46) GO TO 100             NITC0050
IF (I.GT.5.AND.I.LT.23) GO TO 100             NITC0060
I1=I-1                                         NITC0070
HABS=C11(I1)                                    NITC0080
100 CONTINUE                                     NITC0090
TX(11)=HABS*W(11)                            NITC0100
SUM11=TX(11)                                    NITC0110
IF (TX(11).EQ.0.0) GO TO 300                 NITC0120
IF (TX(11).LE.0.1) GO TO 200                 NITC0130
IF (TX(11).GT.20.) GO TO 400                 NITC0140
TX(11)=EXP(-TX(11))                           NITC0150
200 TX(11)=1.0-TX(11)+0.5*TX(11)*TX(11)     NITC0160
GO TO 500                                       NITC0170
300 TX(11)=1.0                                 NITC0180
GO TO 500                                       NITC0190
400 TX(11)=0.0                                 NITC0200
500 RETURN                                      NITC0220
END                                            NITC0230

```

```

SUBROUTINE NITRO(I,W,C4,TX,SUM4)          NITR0010
DIMENSION C4(33),TX(4),W(4)                NITR0020
C*****TRANSMITTANCE FOR NITROGEN CONTINUUM*****NITR0030
IF (I.LT.64) GO TO 200                     NITR0040
I1=I-63                                     NITR0050
C     TEMP FIX FOLLOWS                     NITR0060
C     IF (I1.GT.10) GO TO 300                 NITR0070
TX(4)=C4(I1)*W(4)                         NITR0080
SUM4=TX(4)                                  NITR0090
IF (TX(4).EQ.0.0) GO TO 200                 NITR0100
IF (TX(4).LE.0.1) GO TO 100                 NITR0110
IF (TX(4).GT.20.) GO TO 300                 NITR0120
TX(4)=EXP(-TX(4))                          NITR0130
GO TO 400                                     NITR0140
100 TX(4)=1.0-TX(4)+0.5*TX(4)*TX(4)        NITR0150
GO TO 400                                     NITR0160
200 TX(4)=1.0                                 NITR0170
GO TO 400                                     NITR0180
300 TX(4)=0.0                                 NITR0190
400 RETURN                                    NITR0200
END                                         NITR0210

```

```

SUBROUTINE ND2(I,W,C14,TX) NO2X0010
COMMON /M03/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FN02(9), NO2X0020
  O1(9),O2(9),PPMS02,PPMNH3,PPMN02 NO2X0030
  DIMENSION C14(6),TX(14),W(14) NO2X0040
C***** NO2X0050
C THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY ND2 ( PPM READ IN NO2X0060
C THE MAIN PROGRAM). NO2X0070
C***** NO2X0080
C***** NO2X0090
C***** NO2X0100
  IF (I.GE.102.AND.I.LE.104) I1=I-98 NO2X0110
  IF (I.LE.3) I1=1 NO2X0120
  IF (W(14).LT.1.0E-20) GO TO 3 NO2X0130
  WS14=ALOG10(W(14))+C14(I1) NO2X0140
  DO 1 J=1,9 NO2X0150
  IF (WS14-FN02(J)).GT.2.2,1 NO2X0160
1  CONTINUE NO2X0170
2  TX(14)=EXP(-10**((O1(J)+O2(J)*WS14)) NO2X0180
3  RETURN NO2X0190
END NO2X0200

```

```
SUBROUTINE OZONE(I,W,C3,TX)          OZN00010
DIMENSION C3(86),TX(3),W(3)          OZN00020
C*****TRANSMITTANCE FOR OZONE*****
IF (W(3).LT.1.0E-20) GO TO 500    OZN00030
IF (I.LE.22) I1=I                   OZN00040
IF (I.GE.60) I1=I-37                OZN00050
WS3=ALOG10(W(3))+C3(I1)            OZN00060
TX(3)=1/(1+EXP(-3.08019+2.11127*WS3)) OZN00070
500 RETURN                          OZN00080
END                                OZN00090
                                    OZN00100
```

```

SUBROUTINE POINT (X,N,NP,TX)          POI00010
COMMON /M01/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL    POI00020
COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1      POI00030
COMMON /IOUNIT/IOUNT,IOUT,IPHFUN,LOUNT,NDIRTU,NCLIMT,KSTOR,NPLOTUPOI00040
COMMON /SPOTLO/ISPOT,LOREAD,N16          POI00050
DIMENSION TX(16)                      POI00060
LOGICAL ISPOT,N16,LOREAD             POI00070
C*****SUBROUTINE POINT INTERPOLATES EXPONENTIALLY TO          POI00080
C DETERMINE THE EQUIVALENT ABSORBER AMOUNTS AT THAT ALTITUDE.  POI00090
C
C X IS THE HEIGHT IN QUESTION          POI00100
C N IS THE LEVEL INTEGER CORRESPONDING TO X OR THE LEVEL BELOW X  POI00110
C NP =1 IF X COINCIDES WITH MODEL ATMOSPHERE LEVEL ,IF NOT NP = 0  POI00120
C TX(1-8) ARE ABSORBER AMOUNTS PER KM AT HEIGHT X  POI00130
C
C*****555 N=NL          POI00140
C      NP=0          POI00150
C      IF (X.LT.0.0) X=0.          POI00160
C      IF (X.GT.Z(NL)) GO TO 400  POI00170
C      DO 100 I=1,NL            POI00180
C      N=I          POI00190
C      IF (X-Z(I)) 200,400,100  POI00200
C100  CONTINUE          POI00210
C200  J2=N          POI00220
C      N=N-1          POI00230
C      FAC=(X-Z(N))/(Z(J2)-Z(N))  POI00240
C      DO 300 K=1,KMAX          POI00250
C      IF (K.EQ.9.OR.K.EQ.7) GO TO 300  POI00260
C      TX(K)=0.0          POI00270
C      IF (EH(K,N).EQ.0.0) GO TO 300  POI00280
C      IF (EH(K,N).GT.1000.0) GO TO 300  POI00290
C      TX(K)=EH(K,N)*(EH(K,J2)/EH(K,N))**FAC  POI00300
C300  CONTINUE          POI00310
C      GO TO 700          POI00320
C400  NP=1          POI00330
C      DO 500 K=1,KMAX          POI00340
C500  TX(K)=EH(K,N)          POI00350
C700  RETURN          POI00360
CEND                         POI00370

```

```

FUNCTION RESFN (NR,WAVE)          RES00010
THIS FUNCTION WILL READ IN UP TO 20 VALUES OF A RESPONSE FUNCTION RES00020
IF THE RESF CARD IS READ IN EOMAIN. ONLY ONE RESPONSE FUNCTION RES00030
PER RUN IS ALLOWED. THIS FUNCTION WILL ALSO DO A LINEAR INTERPOLATION RES00040
OVER WAVELENGTH. IF RESF CARD IS NOT READ A VALUE OF 1 IS RETURNED RES00050
TO THE CALLING PROGRAM WHEN THIS FUNCTION IS REFERENCED. RES00060
COMMON /IDUNIT/I0IN,I0OUT,IPHFUN,L0UNIT,NDIRTU,NCLIMT,K$TOR,NPLOTURE RES00070
DIMENSION WAVELN(20),RESPFN(20)             RES00080
DATA WAVELN,RESPFN,ICOUNT,NBR /20*0.,20*0.,0,1/           RES00090
IF (NR.NE.1) GO TO 6                   RES00100
ICOUNT=ICOUNT+1                      RES00110
IF (ICOUNT.GT.1) GO TO 2              RES00120
READ (I0IN,100) NBR                  RES00130
IF (NBR.GT.20) WRITE (I0OUT,102)      RES00140
IF (NBR.GT.20) STOP                  RES00150
WRITE (I0OUT,103)                   RES00160
DO 3 I=1,NBR                       RES00170
READ (I0IN,101) WAVELN(I),RESPFN(I)   RES00180
WRITE (I0OUT,104) WAVELN(I),RESPFN(I)  RES00190
IF (WAVE.LT.(WAVELN(1)-.0001).OR.WAVE.GT.(WAVELN(NBR)+.0001)) RES00200
+ GO TO 6                           RES00210
DO 4 I=1,NBR                       RES00220
K=I                               RES00230
IF (WAVE.GE.WAVELN(I)) GO TO 5      RES00240
IF (WAVE/WAVELN(K).GE..99.AND.WAVE/WAVELN(K).LE.1.01) GO TO 7  RES00250
IF (K.EQ.NBR) GO TO 8               RES00260
RESFN=(WAVE-WAVELN(K))*(RESPFN(K+1)-RESPFN(K))/ RES00270
1 (RESPFN(K+1)-RESPFN(K))+RESPFN(K)  RES00280
RETURN                            RES00290
RESFN=RESPFN(K)                  RES00300
RETURN                            RES00310
RESFN=RESPFN(NBR)                RES00320
RETURN                            RES00330
RESFN=1.                          RES00340
RETURN                            RES00350
100  FORMAT (I2)                   RES00360
101  FORMAT (2(E10.4,1X))          RES00370
102  FORMAT (1H ,51HTHE NUMBER OF VALUES FOR THE RESPONSE FUNCTION IS GRE00380
+ ,58HGREATER THAN THE DIMENSIONS LIMITS OF WAVELN() AND RESPFN(), RES00390
+ ,1X,19HPROGRAM TERMINATED.)        RES00400
103  FORMAT (1H ,20X,23HINPUT RESPONSE FUNCTION,/ ,1X,15X,10HWAVELENGTH,RES00410
+ 5X,10HR FUNCTION)               RES00420
104  FORMAT (1H ,15X,2(E10.4,1X))  RES00430
END                                RES00440

```

```

SUBROUTINE S02(I,W,C12,TX)                               S02X0010
COMMON /M03/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FN02(9),   S02X0020
  O1(9),O2(9),PPMS02,PPMNH3,PPMN02                      S02X0030
DIMENSION C12(15),TX(12),W(12)                         S02X0040
C*****THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY S02 ( PPM READ IN S02X0050
THE MAIN PROGRAM).                                     S02X0060
C*****                                                       S02X0070
IF (W(12).LT.1.0E-20) GO TO 5                          S02X0080
IF (I.GE.13.AND.I.LE.22) I1=I-12                      S02X0090
IF (I.GE.82) I1=I-71                                    S02X0100
WS12=ALOG10(W(12))+C12(I1)                           S02X0110
DO 1 J=1,9                                              S02X0120
IF(WS12-FS(J)) 2,2,1                                  S02X0130
CONTINUE                                              S02X0140
TX(12)=EXP(-10**(S1(J)+S2(J)*WS12))                 S02X0150
RETURN                                                 S02X0160
END                                                   S02X0170
1

```

```
SUBROUTINE UNIMIX(I,W,C2,TX)          UNI00010
DIMENSION C2(258),TX(2),W(2)          UNI00020
C*****TRANSMITTANCE FOR UNIFORMLY MIXED GASES*****UNI00030
IF (W(2).LT.1.0E-20) GO TO 500        UNI00040
IF (I.LE.22) I1=I                      UNI00050
IF (I.GE.60.AND.I.LE.126) I1=I-37      UNI00060
IF(I.GE.210.AND.I.LE.363) I1=I-120    UNI00070
IF (I.GE.607) I1=I-363                 UNI00080
WS2=ALOG10(W(2))+C2(I1)                UNI00090
TX(2)=EXP(-10**(-1.14619+0.55013*WS2)) UNI00100
500 RETURN                               UNI00110
END                                     UNI00120
```

```

SUBROUTINE UVÖZNE(I,W,C8,TX,SUM8)           UVZ00010
DIMENSION C8(82),TX(8),W(8)                  UVZ00020
C*****+ TRANSMITTANCE FOR UV OZONE *****UVZ00030
AI=I                                         UVZ00040
IF(I.LE.1159) GO TO 90                      UVZ00050
IF(I.GE.1335) GO TO 100                      UVZ00060
90 XX=10.0                                     UVZ00070
XI=(AI-610.0)/XX+1.0                         UVZ00080
L1=1                                         UVZ00090
L2=53                                        UVZ00100
GO TO 200                                     UVZ00110
100 XX=25.0                                     UVZ00120
XI=(AI-1335.0)/XX+57.0                        UVZ00130
L1=57                                         UVZ00140
L2=102                                        UVZ00150
200 DO 300 N=L1,L2                           UVZ00160
XD=XI-FLOAT(N)                                UVZ00170
IF(XD) 500,400,300                           UVZ00180
300 CONTINUE                                    UVZ00190
400 TX(8)=W(8)*C8(N)                         UVZ00200
GO TO 600                                     UVZ00210
500 TX(8)=C8(N)+XD*(C8(N)-C8(N-1))          UVZ00220
TX(8)=W(8)*TX(8)                            UVZ00230
600 SUM8=TX(8)                                 UVZ00240
IF(TX(8).EQ.0.0) GO TO 800                  UVZ00250
IF(TX(8).LE.0.1) GO TO 700                  UVZ00260
IF(TX(8).GT.20.0) GO TO 900                  UVZ00270
TX(8)=EXP(-TX(8))                           UVZ00280
GO TO 1000                                    UVZ00290
700 TX(8)=1.0-TX(8)+0.5*TX(8)*TX(8)          UVZ00300
GO TO 1000                                    UVZ00310
800 TX(8)=1.0                                 UVZ00320
GO TO 1000                                    UVZ00330
900 TX(8)=0.0                                 UVZ00340
1000 RETURN                                   UVZ00350
END                                         UVZ00360

```

SUBROUTINE SPOT(WAVN1,WAVN2,VIS,NR,IERR,MULDV)

 INPUT: EXCLUDING THE OPTIONAL RESPONSE FUNCTION CARDS,
 THERE IS A MAXIMUM OF 7 CARDS TO EXECUTE THIS MODULE.
 THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OF
 THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.
 THE CARDS ARE INPUT WITH FORMAT (A4,6X,7E10.4)
 EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4
 FOLLOWED BY AS MANY <REAL> FIELDS AS NEEDED, 10 COL PER
 FIELD BEGINNING IN COL 11.
 THE CARDS ARE NOT ORDER DEPENDENT.
 SPOT0010 SPOT0020
 SPOT0030 SPOT0040
 SPOT0050 SPOT0060
 SPOT0070 SPOT0080
 SPOT0090 SPOT0100
 SPOT0110 SPOT0120
 SPOT0130 SPOT0140
 SPOT0150 SPOT0160
 SPOT0170 SPOT0180
 SPOT0190 SPOT0200
 SPOT0210 SPOT0220
 SPOT0230 SPOT0240
 SPOT0250 SPOT0260
 SPOT0270 SPOT0280
 SPOT0290 SPOT0300
 SPOT0310 SPOT0320
 SPOT0330 SPOT0340
 SPOT0350 SPOT0360
 SPOT0370 SPOT0380
 SPOT0390 SPOT0400
 SPOT0410 SPOT0420
 SPOT0430 SPOT0440
 SPOT0450 SPOT0460
 SPOT0470 SPOT0480
 SPOT0490 SPOT0500
 SPOT0510 SPOT0520
 SPOT0530 SPOT0540
 SPOT0550 SPOT0560
 SPOT0570 SPOT0580
 SPOT0590 SPOT0600
 SPOT0610 SPOT0620
 SPOT0630 SPOT0640
 SPOT0650 SPOT0660
 SPOT0670 SPOT0680
 SPOT0690 SPOT0700

CARD 1
 ENVR ISORC, ITARG, IHAZE, MODEL, NLAM
 ISORC = 0 SUNLIGHT ONLY SPOT0130
 1 MOONLIGHT ONLY SPOT0140
 2 EMISSION ONLY SPOT0150
 3 SUNLIGHT AND EMISSION SPOT0160
 4 MOONLIGHT AND EMISSION SPOT0170
 ITARG = 0 BACKGROUND ONLY SPOT0180
 1 GROUND REFLECTANCE/EMISSION SPOT0190
 2 TARGET REFLECTANCE/EMISSION SPOT0200
 ** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK **
 FOR SLANT PATHS IHAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES. SPOT0210
 IHAZE = 0, NO AEROSOL ATTENUATION SPOT0220
 =1, MARITIME POLAR SPOT0230
 =2, MARITIME ARCTIC SPOT0240
 =3, CONTINENTAL POLAR SPOT0250
 =4, RAIN SPOT0260
 =5, SNOW SPOT0270
 =7, USER SUPPLIED EXTINCTION COEFFICIENT SPOT0280
 <READ ON ATM CARD - SEE CARD 3 BELOW> SPOT0290
 =8, EXTINCTION COEFFICIENT WILL BE READ FROM SPOT0300
 PHASE FUNCTION DATA FILE SPOT0310
 MODEL = 1 TROPICAL MODEL ATMOSPHERE SPOT0320
 2 MIDLATITUDE SUMMER SPOT0330
 3 MIDLATITUDE WINTER SPOT0340
 4 SUBARCTIC SUMMER SPOT0350
 5 SUBARCTIC WINTER SPOT0360
 6 1962 US STANDARD SPOT0370
 8 ISRAELI STANDARD (YEAR, DAYTIME) SPOT0380
 9 ISRAELI STANDARD (YEAR, NIGHTTIME) SPOT0390
 NLAM OPTION FOR AEROSOL PHASE FUNCTION SPOT0400
 = 0 NO AEROSOL ATTENUATION SPOT0410
 NE 0 READ PHASE FUNCTION DATA SET - ALSO SEE SPOT0420
 ID BELOW AND EXPLN OF PFN DATA SET BELOW SPOT0430
 ID PHASE FUNCTION IDENTIFIER SPOT0440
 =0, USER SUPPLIED SPOT0450
 =1, MARITIME ARCTIC, VIS=0.1 TO 2.0 KM SPOT0460
 =2, MARITIME POLAR, VIS=0.2 KM SPOT0470
 =3, MARITIME POLAR, VIS=0.2 KM SPOT0480
 =4, CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM SPOT0490
 =5, WHITE PHOSPHORUS SPOT0500
 =6, HEXACHLOROETHANE SPOT0510
 =7, FOG OIL SPOT0520
 =8, DUST (MODERATE AEROSOL LOADING) SPOT0530
 =9, DUST (HEAVY AEROSOL LOADING) SPOT0540
 =10, MARITIME MODEL B, VIS=5KM, RH=95% SPOT0550
 =11, MARITIME MODEL B, VIS=10KM, RH=90% SPOT0560
 =12, MARITIME MODEL B, VIS=50KM, RH=50% SPOT0570
 EMIS CARD 2 **** IF ISORC LT 2 OR ITARG LT 1 THIS CARD IS NOT NEEDED SPOT0580
 EM(1), TM(1), EM(2), TM(2) SPOT0590
 EMC(1) EMISSIVITY OF GROUND SPOT0600
 TM(1) TEMPERATURE OF GROUND (KELVIN) SPOT0610
 EMC(2) EMISSIVITY OF TARGET SPOT0620
 TM(2) TEMPERATURE OF TARGET (KELVIN) SPOT0630
 ATM CARD 3 ZENTH, CLDHGT, PHASE,BETAEX SPOT0640
 ZENTH INCIDENT ANGLE OF SUNLIGHT OR MOONLIGHT (DEGREES) SPOT0650
 CLDHGT HEIGHT OF BOTTOM OF CLOUD LAYER (KM) SPOT0660
 ONLY NEEDED WHEN IHAZE NE 0 (DEFAULT IS 0.) SPOT0670

PHASE	PHASE ANGLE FOR MOONLIGHT (DEGREES)	SPOT0710	
BETAEX	USER SUPPLIED EXTINCTION COEFFICIENT	SPOT0720	
	VALID ONLY WHEN IHAZE=7	SPOT0730	
CARD 4			
TARG	RTARG,COSX,COSY,COSZ	SPOT0740	
RTARG	SLANT RANGE FROM RECEIVER TO TARGET (KM)	SPOT0750	
COSX	X-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)	SPOT0760	
COSY	Y-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)	SPOT0770	
COSZ	Z-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)	SPOT0780	
CARD 5 **** IF ITARG LT 1 THIS CARD IS NOT NEEDED			SPOT0790
REFL	A0(1),A1(1),IALB(1),A0(2),A1(2),IALB(2)	SPOT0800	
A0(1)	ALBEDO COEFFICIENT FOR GROUND	SPOT0810	
A1(1)	ALBEDO COEFFICIENT FOR GROUND	SPOT0820	
IALB(1)	TYPE OF REFLECTION DISTRIBUTION FOR GROUND	SPOT0830	
A0(2)	ALBEDO COEFFICIENT FOR TARGET	SPOT0840	
A1(2)	ALBEDO COEFFICIENT FOR TARGET	SPOT0850	
IALB(2)	TYPE OF REFLECTION DISTRIBUTION FOR TARGET	SPOT0860	
IALB = 0	LAMBERTIAN REFLECTION SURFACE	SPOT0870	
	1 ISOTROPIC	SPOT0880	
		SPOT0890	
CARD 6			SPOT0900
SENS	ALT, THETA, PHI, SANG2	SPOT0910	
ALT	ALTITUDE OF RECEIVER (KM)	SPOT0920	
THETA	POLAR DIRECTION OF LOOK ANGLE (DEGREES)	SPOT0930	
PHI	AZIMUTH DIRECTION OF LOOK ANGLE (DEGREES)	SPOT0940	
*** METEOROLOGICAL AZIMUTH CONVENTION ASSUMED: N = 0 DEG,			SPOT0950
*** E = 90 DEG, S = 180 DEG, W = 270 DEG *****			SPOT0960
SANG2 HALF ANGLE DEFINING RECEIVER FIELD-OF-VIEW			SPOT0970
(DEGREES)			SPOT0980
CARD 7			SPOT1000
GO	SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FOR THIS CALL. NOTE THAT IF A DATA CARD IS NOT READ THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLS TO THE MODULE ARE STILL IN EFFECT.	SPOT1010	
OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY NR=1 IN EOMAIN) THESE CARDS MUST FOLLOW THE GO CARD AND CAN ONLY BE INSERTED ONCE			SPOT1020
CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2).			SPOT1030
CARDS 2 - NUMBER OF VALUES; FORMAT (2(E10.4,1X)) ONE VALUE OF WAVE (UM) AND RESPONSE FUNCTON PER CARD			SPOT1040
N.B. ONLY ONE RESPONSE FUNCTION PER EOSAEL RUN.			SPOT1050
AUXILLARY READ FROM UNIT IPHFUN (ASL DATA SET PROVIDED WITH EOSAEL)			SPOT1060
ANG	ANGLES AT WHICH PHASE MATRIX IS DEFINED	SPOT1070	
NANG	NANG VALUES (DEFAULT IS 65), FORMAT(11(F6.2,1X))	SPOT1080	
NANG, ID, WAVE, OMEGA0, BETAEX, BETABS	SPOT1090		
PF	NUMBER OF ANGLES AT WHICH THE PHASE FUNCTION HAS VALUES, PFN IDENTIFIER, WAVELENGTH(UM), SINGLE SCATTERING ALBEDO, EXTINCTION COEFFICIENTS (TOTAL AND SCATTERING), FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X)).	SPOT1100	
	FORMAT (6(E12.6,1X))	SPOT1110	
		SPOT1120	
*****			SPOT1130
LOGICAL L1,L2,L3,L4,L5,L6,L7,ISPOT,N16,LOREAD,HORIZ	SPOT1140		
DIMENSION DUMMY(16)	SPOT1150		
EQUIVALENCE (ITARG,IT)	SPOT1160		
COMMON /ANSW2/TTR(16),TBR(16),CNTRST(16)	SPOT1170		
COMMON /ALBEDO/A0(2),A1(2),IALB(2)	SPOT1180		
COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB	SPOT1190		
COMMON /CGEOM/COSGM,COSBT,COSIN	SPOT1200		
COMMON /COM11/ISORC,ITARG,IWN,JHL	SPOT1210		
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK	SPOT1220		
COMMON /CTARG/RTARG,COSX,COSY,COSZ	SPOT1230		
COMMON /EMISS/EM(2),TM(2)	SPOT1240		
COMMON /IOUNIT/I0IN,I0OUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSPOT1250	SPOT1250		
COMMON /M01/EHC(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,	SPOT1260		
+ RRS(16,34),SCDE(16,34),	SPOT1270		
+ TRANS(16,5),RADA(16,2),WAVE(16),SS(16),	SPOT1280		
		SPOT1290	
		SPOT1300	
		SPOT1310	
		SPOT1320	
		SPOT1330	
		SPOT1340	
		SPOT1350	
		SPOT1360	
		SPOT1370	
		SPOT1380	
		SPOT1390	
		SPOT1400	

```

1      DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),          SPOT1410
2      PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)        SPOT1420
COMMON /MO2/ WO(34),RO,TBOUND,JP,IM,ML,IP,JSTOR           SPOT1430
COMMON /EM2/WC(16),E(16),IL,IKMAX,LENTOR,NLL            SPOT1440
COMMON /SPOTLO/ ISPOT,LREAD,N16                          SPOT1450
COMMON /LOWEX/ WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX, SPOT1460
1 CLDHGT,NCLD                                         SPOT1470
COMMON /BASBOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16), SPOT1480
1 BE(16),SINGWV,PFC(65),LMAX                         SPOT1490
COMMON /LOGIC/L1,L2,L3,L4,L5,L6,L7                   SPOT1500
DATA L1,L2,L3,L4,L5,L6,L7/7*.FALSE./               SPOT1510
DATA ITR1,ITR2,ITR3,ITR4/2,3,5,1/                  SPOT1520
N16=.TRUE.                                           SPOT1530
INBP=16                                              SPOT1540
DUM=1                                                SPOT1550
ICLMAT=0                                             SPOT1560
LREAD=.TRUE.                                         SPOT1570
ISPOT=.TRUE.                                         SPOT1580
C INITIALIZE AND READ INPUT PARAMETERS             SPOT1590
CALL ZERO                                         SPOT1600
CALL INDAT(IEMISS,IHAZE,IM,LEN,ML,MODEL,          SPOT1610
1     TBOUND,0,CLDHGT,BETAEX)                      SPOT1620
C CHECK FOR ERROR IN INPUT DATA                  SPOT1630
IF (IHAZE.EQ.1) IERR=1                           SPOT1640
IF (IERR.EQ.1) RETURN                            SPOT1650
C FIRST CALL IS TO READ LOWTRAN DATA FILE ONLY   SPOT1660
FIRST CALL IS TO READ LOWTRAN DATA FILE ONLY       SPOT1670
CALL LT4M(ALT,DUM,ZENTH,3,0,TRANS(1,1),DUMMY,DUMMY, SPOT1680
1     IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,          SPOT1690
2     ICLMAT,IERR,NR,IHAZE,MULDV)                 SPOT1700
V1=WAVN1                                         SPOT1710
V2=WAVN2                                         SPOT1720
CALL CKER(V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,DUM) SPOT1730
WAVE<1>=10000./V1                                SPOT1740
DO 300 IW=2,INBR                                  SPOT1750
V2=V1+20.*FLOAT(MULDV)*FLOAT(IW-1)              SPOT1760
IF (V2.GE.WAVN2) GO TO 400                      SPOT1770
WAVE<IW>=10000./V2                                SPOT1780
300 CONTINUE                                         SPOT1790
L1=.TRUE.                                         SPOT1800
C MAXIMUM NO. OF WAVELENGTHS                    SPOT1810
400 IW=IW-1                                      SPOT1820
IF (L1) IW=INBR                                  SPOT1830
NWVL=IWN                                         SPOT1840
CCCCC ARRAY WVL IS USED ONLY IN SUBROUTINE PFUNC. THE WAVELENGTHS   SPOT1850
IN THIS ARRAY INCREASE WITH INCREASING ARRAY INDEX. THE          SPOT1860
VALID RANGES FOR VALUES IN THE WVL ARRAY ARE : 0.2-2.0, 3.0-5.0,   SPOT1870
AND 8.0-12.0 MICROMETERS.                               SPOT1880
SPOT1890
DO 355 JX=1,NWVL                                  SPOT1900
INDM=NWVL-JX+1                                    SPOT1910
355 WVL(JX)=WAVE<INDM>                           SPOT1920
DO 500 I=1,NL                                     SPOT1930
IF (ALT.LE.Z(I)) GO TO 600                      SPOT1940
500 CONTINUE                                         SPOT1950
WRITE (I00UT,3700) ALT,I,Z(I)                   SPOT1960
IERR=1                                            SPOT1970
RETURN                                            SPOT1980
600 CONTINUE                                         SPOT1990
IF (I.EQ.1) WRITE (I00UT,3800) ALT              SPOT2000
IF (I.EQ.1) IERR=1                               SPOT2010
IF (IERR.EQ.1) RETURN                            SPOT2020
JHL=I-1                                           SPOT2030
NLL=NL-1                                           SPOT2040
SANG=TWOPI*(1.0-COS(SANG2*PIRAD))              SPOT2050
CZNTH=COS(ZENTH*PIRAD)                          SPOT2060
SZNTH=SIN(ZENTH*PIRAD)                          SPOT2070
CTHTA=COS(THETA*PIRAD)                          SPOT2080
STHTA=SIN(THETA*PIRAD)                          SPOT2090
STHTA=SIN(THETA*PIRAD)                          SPOT2100

```

```

COPHI=COS(PHI*PIRAD)
SPHIF=SIN(PHI*PIRAD)
IF (ISORC.EQ.2) GO TO 900
C CALCULATE SOURCE TERM ... FOR SUNLIGHT (ISORC=0,3)
C OR MOONLIGHT (ISORC=1,4) SPOT2110
DO 700 IW=1,IWN SPOT2120
SS(IW)=0.0 SPOT2130
IF (ISORC.EQ.0.OR.ISORC.EQ.3) SS(IW)=SOLARS(WAVE(IW))
IF (ISORC.EQ.1.OR.ISORC.EQ.4) SS(IW)=SMOON(WAVE(IW),PHASE)
700 CONTINUE SPOT2140
C CALCULATE DIRECT INTENSITY ... FOR SUNLIGHT (ISORC=0,3)
C OR MOONLIGHT (ISORC=1,4) SPOT2150
COSIN=STHTA*COPHI*SZNTH+CTHTA*CZNTN SPOT2160
ANGIN=ACOS(COSIN)/PIRAD SPOT2170
C READ PHASE FUNCTION FILE SPOT2180
CALL INDAT(IEMISS,IAZ,IM,LEN,ML,MODEL, SPOT2190
1 TBOUND,1,CLDHGT,BETAEX) SPOT2200
IF (ANGIN.GT.SANG2) L2=.TRUE. SPOT2210
IF (ZENTH.GT.80.0) L3=.TRUE. SPOT2220
CALL LT4M(ALT,DUM,ZENTH,3,0,TRANS(1,1),DUMMY,DUMMY, SPOT2230
1 IEMISS,LEN,MODEL,VIS, V1,V2,TGRD, SPOT2240
2 ICLMAT,IERR,NR,IAZ,MULDV) SPOT2250
IF (L2.OR.L3) GO TO 900 SPOT2260
DO 800 IW=1,IWN SPOT2270
DIR(IW)=SS(IW)*TRANS(IW,1) SPOT2280
800 CONTINUE SPOT2290
900 IF (ITARG.EQ.0.AND.THETA.GT.90.0) L7=.TRUE. SPOT2300
IF (L7) GO TO 3500 SPOT2310
IF (ITARG.EQ.0) GO TO 1200 SPOT2320
IF (ITARG.EQ.1) GO TO 1000 SPOT2330
C TARGET. SPOT2340
ZTARG=RTARG*CTHTA+ALT SPOT2350
COSTX=STHTA*COPHI SPOT2360
COSTY=STHTA*SPHI SPOT2370
COSTZ=CTHTA SPOT2380
COSBT=COSX*SZNTH+COSZ*CZNTN SPOT2390
COSGM=-COSTX*COSX+COSTY*COSY+COSTZ*COSZ SPOT2400
IF (THETA.LE.90.0) GO TO 1100 SPOT2410
COSBTG=CZNTN SPOT2420
COSGMG=COS((180.0-THETA)*PIRAD) SPOT2430
GO TO 1100 SPOT2440
C GROUND... SPOT2450
1000 ZTARG=0.0 SPOT2460
COSBT=CZNTN SPOT2470
COSGM=COS((180.0-THETA)*PIRAD) SPOT2480
COSBTG=COSBT SPOT2490
COSGMG=COSGM SPOT2500
C CALCULATE ATMOSPHERIC TRANSMISSION/RADIANCE FOR VARIOUS PATHS SPOT2510
1100 IF (COSGM.LE.0.0) L4=.TRUE. SPOT2520
IF (COSBT.LE.0.0) L5=.TRUE. SPOT2530
IF (THETA.LE.90.0.AND.ITARG.EQ.1) L6=.TRUE. SPOT2540
IF (L4.OR.L6) GO TO 3500 SPOT2550
1200 CONTINUE SPOT2560
IF (THETA.EQ.90.0) SPOT2570
1 CALL LT4M(ALT,DUM,1000.0,1,2,TRANS(1,4),RADA(1,1), SPOT2580
2 DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2, SPOT2590
3 TGFD,ICLMAT,IERR,NR,IAZ,MULDV) SPOT2600
IF (THETA.EQ.90.0.AND.ITARG.EQ.2) SPOT2610
1 CALL LT4M(ALT,DUM,RTARG,1,2,TRANS(1,2),RADA(1,2), SPOT2620
2 DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2, SPOT2630
3 TGFD,ICLMAT,IERR,NR,IAZ,MULDV) SPOT2640
IF (THETA.EQ.90.0) GO TO 1300 SPOT2650
1 CALL LT4M(ALT,DUM,THETA,3,2,TRANS(1,4),RADA(1,2),DUMMY, SPOT2660
2 IEMISS,LEN,MODEL,VIS, V1,V2,TGRD, SPOT2670
3 ICLMAT,IERR,NR,IAZ,MULDV) SPOT2680
IF (ITARG.EQ.0) GO TO 2000 SPOT2690
IF (THETA.LT.90.0) SPOT2700
1 CALL LT4M(ALT,ZTARG,THETA,2,2,TRANS(1,2),RADA(1,2), SPOT2710
2 DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2, SPOT2720

```

```

3      TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)           SPOT2810
IF (<THETA,LT,.90.0) GO TO 1300                 SPOT2820
CALL LT4M(ZTARG,THETA,2,2,TRANS(1,2),RADAC(1,2),    SPOT2830
1      RADG(1))                                     IEMISS,LEN,MODEL,VIS, V1,V2,
2      TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)           SPOT2840
IF (<ITARG,NE,2) ITR4=2                         SPOT2850
IF (<ITARG,NE,2) GO TO 1300                     SPOT2860
IF (<ZTARG,LE,0,0) ITR4=2                       SPOT2870
IF (<ZTARG,LE,0,0) GO TO 1300                   SPOT2880
ITR1=4                                         SPOT2890
CALL LT4M(ALT,0,0,THETA,2,2,TRANS(1,4),RADAC(1,1),   SPOT2900
1      RADG(1))                                     IEMISS,LEN,MODEL,VIS, V1,V2,
2      TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)           SPOT2910
1300 IF (<ISORC,LT,2) GO TO 1600                SPOT2920
C      CALCULATE UNCOLLIDED EMISSION ... FROM GROUND (ITARG=1)   SPOT2930
C      OR TARGET (ITARG=2)                           SPOT2940
C      DO 1500 IW=1,IWN                          SPOT2950
C      WAVEM=WAVE(IW)/1.0E+4                      SPOT2960
C      IF (<ITARG,EQ,2) GO TO 1400                SPOT2970
C      BKG(IW)=BLACK(WAVEM,TGRD)*EM(1)            SPOT2980
C      RADG(IW)=RADG(IW)*EM(1)*COSGMG          SPOT2990
C      BK(IW)=BKG(IW)                            SPOT3000
C      UTEM(IW)=RADG(IW)                         SPOT3010
C      GO TO 1500                                SPOT3020
1400 BK(IW)=BLACK(WAVEM,TM(2))*EM(2)             SPOT3030
UTEM(IW)=BK(IW)*COSGM*TRANS(IW,2)               SPOT3040
IF (<THETA,LE,90.0) GO TO 1500                  SPOT3050
BK(IW)=BLACK(WAVEM,TGRD)*EM(1)                  SPOT3060
RADG(IW)=RADG(IW)*EM(1)*COSGMG                SPOT3070
1500 CONTINUE                                    SPOT3080
1600 IF (<ISORC,EQ,2) GO TO 2900                SPOT3090
C      CALCULATE UNCOLLIDED REFLECTANCE ... FROM GROUND (ITARG=1)   SPOT3100
C      OR TARGET (ITARG=2)                           SPOT3110
C      IEMISS=0                                     SPOT3120
C      IF (<L5) GO TO 2000                         SPOT3130
C      HORIZ=ABS(ZTARG-ALT).LT.0.001              SPOT3140
C      IF(HORIZ) ITR2=1                           SPOT3150
C      IF(HORIZ) GO TO 1700                      SPOT3160
C      CALL LT4M(ZTARG,DUM,ZENTH,3,2,TRANS(1,3),DUMMY,DUMMY,   SPOT3170
1      IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,           SPOT3180
2      ICLMAT,IERR,NR,IHAZE,MULDV)             SPOT3190
1700 IF (<ZTARG,LE,0,0) ITR3=ITR2               SPOT3200
IF (<ZTARG,LE,0,0) GO TO 1800                 SPOT3210
IF (<THETA,LE,90.0) GO TO 1800                 SPOT3220
CALL LT4M(0,0,DUM,ZENTH,3,2,TRANS(1,5),DUMMY,DUMMY,   SPOT3230
1      IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,           SPOT3240
2      ICLMAT,IERR,NR,IHAZE,MULDV)             SPOT3250
1800 DO 1900 IW=1,IWN                          SPOT3260
ALB=ALBEDO(IW)                                SPOT3270
UTRF(IW)=SS(IW)*COSBT*ALB*TRANS(IW,2)*TRANS(IW,ITR2)  SPOT3280
IF (<THETA,LE,90.0) GO TO 1900                  SPOT3290
UERF(IW)=SS(IW)*COSBTG*ALBEDO(1)*TRANS(IW,ITR1)*
1      TRANS(IW,ITR3)                           SPOT3300
1900 CONTINUE                                    SPOT3310
C      CALCULATE SINGLE-SCATTERED PATH RADIANCE ...
C      FROM SUNLIGHT (ISORC=0,3)                  SPOT3320
C      OR MOONLIGHT (ISORC=1,4)                   SPOT3330
2000 CALL COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)  SPOT3340
IF (<IERR,EQ,1) RETURN                         SPOT3350
IF (<ITARG,EQ,0) GO TO 2700                   SPOT3360
CALL PATHRD(CTHTA,ALT,RTARG,1,IHAZE,NR,           SPOT3370
1      IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
DO 2100 IW=1,IWN                               SPOT3380
2100 PATHR2(IW)=SS(IW)*PATHR(IW)                SPOT3390
IF (<ABS(CTHTA).LE.1.0E-3) GO TO 2600          SPOT3400
IF (<CTHTA,LT,0,0) GO TO 2300                  SPOT3410
2200 Z2=NLL                                     SPOT3420
I0=2                                         SPOT3430
RT=(Z(NLL)-ZTARG)/CTHTA                      SPOT3440
GO TO 2800                                    SPOT3450

```

```

2300 IF (ZTARG.GT.0.0) GO TO 2500 SPOT3510
    DO 2400 IW=1,IWN SPOT3520
2400 PATHR(IW)=0.0 SPOT3530
    GO TO 2900 SPOT3540
2500 Z2=0.0 SPOT3550
    IO=1 SPOT3560
    RT=-ZTARG/CTHTA SPOT3570
    GO TO 2800 SPOT3580
2600 Z2=ZTARG SPOT3590
    RT=1000.0 SPOT3600
    IO=3 SPOT3610
    GO TO 2800 SPOT3620
2700 IEMISS=0 SPOT3630
    ZTARG=ALT SPOT3640
    IF (ABS(CTHTA).LE.1.0E-03) GO TO 2600 SPOT3650
    GO TO 2200 SPOT3660
2800 CALL PATHRD(CTHTA,ZTARG,RT,IO,IHAZE,NR, SPOT3670
    IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV) SPOT3680
C   CALCULATE BACKGROUND AND TOTAL INTENSITY, PLUS SPOT3690
C   CONTRAST RATIO SPOT3700
2900 DO 3100 IW=1,IWN SPOT3710
    PATHR(IW)=PATHR2(IW)+SS(IW)*PATHR(IW) SPOT3720
    TTR(IW)=PATHR2(IW)+UTRF(IW)+RADAC(IW,2)+UTEM(IW) SPOT3730
    TBR(IW)=PATHR(IW)+UERF(IW)+RADAC(IW,1TR4)+RADG(IW) SPOT3740
    DIF=TTR(IW)-TBR(IW) SPOT3750
    IF (TBR(IW).GT.0.0) GO TO 3000 SPOT3760
    IF (TTR(IW).EQ.TBR(IW)) CNTRST(IW)=0.0 SPOT3770
    IF (TTR(IW).NE.TBR(IW)) CNTRST(IW)=1.0E30 SPOT3780
    GO TO 3100 SPOT3790
3000 CNTRST(IW)=DIF/TBR(IW) SPOT3800
3100 CONTINUE SPOT3810
C   CALCULATE TOTAL RADIANCES INTEGRATED OVER DETECTOR RESPONSE SPOT3820
D2=DV*0.5 SPOT3830
SUMRPF=0. SPOT3840
DO 3200 IW=1,IWN SPOT3850
    NW=100000./WAVE(IW) SPOT3860
    W2=100000./FLOAT(NW)-DV2 SPOT3870
    W1=100000./FLOAT(NW)+DV2 SPOT3880
    IF (IW.EQ.1) W2=WAVE(1) SPOT3890
    IF (IW.EQ.IWN) W1=WAVE(IWN) SPOT3900
    RESPFN=RESFN(NR,WAVE(IW)) SPOT3910
    SUMRPF=SUMRPF+RESPFN SPOT3920
    DW=(W2-W1)*RESPFN SPOT3930
    TOT(1)=TOT(1)+DW*UTEM(IW) SPOT3940
    TOT(2)=TOT(2)+DW*UTRF(IW) SPOT3950
    TOT(3)=TOT(3)+DW*RADAC(IW,2) SPOT3960
    TOT(4)=TOT(4)+DW*PATHR2(IW) SPOT3970
    TOT(5)=TOT(5)+DW*TTR(IW) SPOT3980
    TOT(6)=TOT(6)+DW*RADG(IW) SPOT3990
    TOT(7)=TOT(7)+DW*UERF(IW) SPOT4000
    TOT(8)=TOT(8)+DW*RADAC(IW,1TR4) SPOT4010
    TOT(9)=TOT(9)+DW*PATHR(IW) SPOT4020
    TOT(10)=TOT(10)+DW*TBR(IW) SPOT4030
3200 TOT(11)=TOT(11)+DW*DIR(IW) SPOT4040
    IF (NR.NE.1) SUMRPF=1. SPOT4050
    DO 3250 I=1,11 SPOT4060
3250 TOT(I)=TOT(I)/SUMRPF SPOT4070
    IF (TOT(10).GT.0.0) GO TO 3300 SPOT4080
    IF (TOT(5).EQ.TOT(10)) TOT(12)=0.0 SPOT4090
    IF (TOT(5).NE.TOT(10)) TOT(12)=1.0E30 SPOT4100
    GO TO 3400 SPOT4110
3300 TOT(12)=(TOT(5)-TOT(10))/TOT(10) SPOT4120
3400 CONTINUE SPOT4130
3500 CALL OUTPUT(MODEL,IHAZE,CLDHGT) SPOT4140
    RETURN SPOT4150
C   3700 FORMAT (1H,11H ALTITUDE <,F10.3,17H> GREATER THAN Z<, SPOT4160
    1           I2,2H>=,F10.3,27H CONTROL RETURNED TO MAIN SPOT4170
    2           10H FROM SPOT,> SPOT4180
3800 FORMAT (1H,11H ALTITUDE <,F10.3,16H> LESS THAN ZERO, SPOT4190
                                         SPOT4200

```

1 37H CONTROL RETURNED TO MAIN FROM SPOT.)
END

SPOT4210
SPOT4220

```
FUNCTION ALBEDO(I)
COMMON /ALBED/A0(2),A1(2),IALB(2)
COMMON /CGEOM/COSGM,COSBT,COSIN
COMMON /CONST/PI,PI2,PIRAD,TWOFI,TORRMB,CDEGK
CALCULATE ALBEDO FOR GROUND (ITARG=1) OR TARGET (ITARG=2)
IALB(I) = 0  LAMBERTIAN REFLECTION SURFACE
          1  ISOTROPIC
          A=A0(I)+A1(I)*COSBT
IF (IALB(I).EQ.0) ALBEDO=A*COSGM/PI
IF (IALB(I).EQ.1) ALBEDO=A/TWOFI
RETURN
END
```

ALB00010
ALB00020
ALB00030
ALB00040
ALB00050
ALB00060
ALB00070
ALB00080
ALB00090
ALB00100
ALB00110
ALB00120

FUNCTION BLACK(W,T)
BLACK(W,T) = PLANCK FUNCTION (UNITS: WATT PER SQUARE METER PER
MICROMETER PER STERADIAN), GIVEN WAVELENGTH W IN CM AND TEMP-
ERATURE T IN K

EXP OVERFLOW PROTECTION

ARG=1.43879/(W*T)
IF(ARG.LT.88.) GO TO 1
BLACK=0.0
RETURN

1 BLACK=1.19106E-12/(W**5*(EXP(ARG)-1.0))
RETURN
END

BLA00010
BLA00020
BLA00030
BLA00040
BLA00050
BLA00060
BLA00070
BLA00080
BLA00090
BLA00100
BLA00110
BLA00120
BLA00130
BLA00140

```

SUBROUTINE COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)      COE00010
COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB          COE00020
COMMON /CTARG/RTARG,COSX,COSY,COSZ                      COE00030
COMMON /MD1/DUMMIES(715),MHOLD,NL,                         COE00040
+           RRS(16,34),SCOE(16,34),                           COE00050
+           TRANS(16,5),RADA(16,2),WAVE(16),SS(16),           COE00060
1           DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),       COE00070
2           PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)    COE00080
COMMON /BASBOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16), COE00090
1 BE(16),SINGWV,PFC(65)                                    COE00100
COMMON /COM1/ISORC,ITARG,IWN,JHL                          COE00110
COMMON /CONST/ PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK            COE00120
DIMENSION P(34),T(34)                                      COE00130
CALCULATE THE WAVELENGTH-DEPENDENT CONSTANT PRESSURE COEFFICIENTS COE00140
C FOR MOLECULAR SCATTERING.                                COE00150
C LOOP OVER LAYERS                                         COE00160
DO 600 I=1,NL                                           COE00170
PS=P(I)/1013.0                                         COE00180
TS=CDEGK/T(I)                                         COE00190
RSCAT=PS*TS                                         COE00200
C LOOP OVER WAVELENGTHS                                     COE00210
DO 600 IW=1,IWN                                         COE00220
RAYS=0.0                                              COE00230
NW=10000./WAVE(IW)                                     COE00240
C RAYLEIGH SCATTERING = 0. FOR WAVELENGTH GT 3.33 UM        COE00250
IF (NW.LT.3000) GO TO 200                               COE00260
WN=FLOAT(NW)                                         COE00270
RAYS=RSCAT*(WN**4)/(9.67578E+18-1.11836E+09*WN**2)    COE00280
200 CONTINUE
AEXT=1.                                                 COE00290
AABS=1.                                                 COE00300
IF (IHAZE.EQ.0.OR.I.NE.(NCLD-1)) GO TO 1             COE00310
EXT55=3.912/VIS                                         COE00320
C UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE COE00340
ZTALT=ZTARG/ALT                                         COE00350
IF (ABS(ZTALT-1.).LT..01) RNG=RTARG                   COE00360
IF ((ZTARG.GT.ALTL).AND.(RTARG.GT..5/COS(THETA*PIRAD))) COE00370
1   RNG=.5/COS(THETA*PIRAD)                            COE00380
IF (ZTARG.LT.RTARG.AND.(RTARG.GT..5/COS((180.-THETA)*PIRAD))) COE00390
1   RNG=.5/COS((180.-THETA)*PIRAD)                     COE00400
1   IF (ITARG.EQ.0.AND.(RTARG.GT..5/COS(THETA*PIRAD))) COE00410
1   RNG=.5/COS(THETA*PIRAD)                            COE00420
ISLANT=1                                                 COE00430
IF (ABS(ZTALT-1.).LT..01) ISLANT=0                    COE00440
C CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL COE00450
CALL XSCALE(WAVE(IW),88.,EXT55,XSTRN,IERR,ISLANT,IHAZE,RNG,THETA) COE00460
IF (IERR.EQ.1) RETURN                                  COE00470
AEXT=-ALOG(XSTRN)/RNG                                 COE00480
C USER SUPPLIED COEFC(IHAZE=7), OR READ FROM PFN DATA FILE(IHAZE=8) COE00490
IF (IHAZE.EQ.7) AEXT=BETAEX                          COE00500
IF (IHAZE.EQ.8) AEXT=BE(IW)                           COE00510
IF (WAVE(IW).LT.2.) AABS=1.                            COE00520
IF (WAVE(IW).GE.3..AND.WAVE(IW).LE.5.) AABS=AEXT*.2 COE00530
IF (WAVE(IW).GE.8..AND.WAVE(IW).LE.12.) AABS=AEXT*.45 COE00540
1 CONTINUE
SCOE(IW,I)=AEXT-AABS+RAYS                            COE00550
C CHECK FOR NO AEROSOL PRESENT                        COE00570
IF(SCOE(IW,I).LT.1.E-20)RRS(IW,I)=1.0               COE00580
C AEROSOL AND RAYLEIGH PRESENT                      COE00590
IF(SCOE(IW,I).GE.1.E-20)RRS(IW,I)=RAYS/SCOE(IW,I) COE00600
C CHECK FOR NO RAYLEIGH SCATTERING                  COE00610
IF (RAYS.LT.1.E-20) RRS(IW,I)=0.0                  COE00620
600 CONTINUE
RETURN
END

```

```

SUBROUTINE DIAG          DIAG0010
  THIS SUBROUTINE PRODUCES DIAGNOSTIC COMMENTARY FOR THE    DIAG0020
  SPOT PROGRAM.                                         DIAG0030
  CALLING SEQUENCE: CALL DIAG                         DIAG0040
  EXTERNAL VARIABLES REQUIRED:                      DIAG0050
    THETA <COMMON BLOCK BKDAT>                     DIAG0060
    HLIGHT,HTYPE <COMMON BLOCK HOLRTH>             DIAG0070
    L1,L2,L3,L4,L5,L6 <COMMON BLOCK LOGIC>          DIAG0080
  LOGICAL L1,L2,L3,L4,L5,L6,L7                      DIAG0090
  COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB   DIAG0100
  COMMON /HOLRTH/                                     HITARG(8,3),HISORC(6,5),
  1   HMODEL(5,6),HMNLTC(3),HSNLTC(3),                DIAG0110
  2   HTRGTC(2),HTYPE(2),HGRND(2)                   DIAG0120
  COMMON /LOGIC/L1,L2,L3,L4,L5,L6,L7               DIAG0130
  COMMON /IOUNIT/IOUT,IIN,IOUNIT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUDIAG0140
  DATA I/1/                                         DIAG0150
  C       WRITE HEADING,                               DIAG0160
  C       WRITE (IOUT,900)                            DIAG0170
  C       IF ERRORS, GO TO 5; OTHERWISE PRINT CLEAN RUN MESSAGE.  DIAG0180
  C       IF (L1.OR.L2.OR.L3.OR.L4.OR.L5.OR.L6.OR.L7) GO TO 100  DIAG0190
  C       WRITE (IOUT,1000)                           DIAG0200
  C       GO TO 700                                  DIAG0210
  C       HERE IF THERE WERE ERRORS                 DIAG0220
  100  WRITE (IOUT,1100)                           DIAG0230
  C       IF (.NOT.L1) GO TO 200                  DIAG0240
  C       WRITE (IOUT,1200) I                      DIAG0250
  C       I=I+1                                    DIAG0260
  200  IF (.NOT.L2) GO TO 300                  DIAG0270
  C       WRITE (IOUT,1300) I,HLIGHT               DIAG0280
  C       I=I+1                                    DIAG0290
  300  IF (.NOT.L3) GO TO 400                  DIAG0300
  C       WRITE (IOUT,1400) I,HLIGHT               DIAG0310
  C       I=I+1                                    DIAG0320
  400  IF (.NOT.L4) GO TO 500                  DIAG0330
  C       WRITE (IOUT,1500) I,HTYPE                DIAG0340
  C       I=I+1                                    DIAG0350
  500  IF (.NOT.L5) GO TO 600                  DIAG0360
  C       WRITE (IOUT,1600) I,HLIGHT               DIAG0370
  C       I=I+1                                    DIAG0380
  600  IF (.NOT.L6) GO TO 700                  DIAG0390
  C       WRITE (IOUT,1700) I,THETA                DIAG0400
  C       I=I+1                                    DIAG0410
  700  IF (.NOT.L7) GO TO 800                  DIAG0420
  C       WRITE (IOUT,1800) I,THETA                DIAG0430
  C       WRITE FOOTING,                         DIAG0440
  C       WRITE (IOUT,1900)                       DIAG0450
  C
  900  FORMAT (1H0,21X,90(1H*),3(/,21X,1H*,88X,1H*))  DIAG0460
  1000 FORMAT (21X,1H*,28X,29HNO SPOT DIAGNOSTICS FOR THIS  DIAG0470
  1     3HRUN,28X,1H*,/,21X,1H*,28X,8H--- ---- ,11(1H-)  DIAG0480
  2     13H --- --- --- 28X,1H*)                  DIAG0490
  1100 FORMAT (21X,1H*,28X,25HSPOT DIAGNOSTIC MESSAGES  DIAG0500
  1     7HFOLLOW,28X,1H*,/,21X,1H*,28X,5H--- ---- ,10(1H-  DIAG0510
  2     ),1X,8(1H-),1X,6(1H-),29X,1H*,2(/,21X,1H*,88X,  DIAG0520
  3     1H*) )                                DIAG0530
  1200 FORMAT (21X,1H*,9X,I1,30H. NUMBER OF WAVELENGTHS (IWN)  DIAG0540
  1     9H EXCEEDS .31 ALLOWABLE DIMENSIONS; IWN RESET  DIAG0550
  2     8X,1H*/,21X,1H*,13X,6HTO 16.,69X,1H*/,,21X,  DIAG0560
  3     1H*,88X,1H*)                                DIAG0570
  1300 FORMAT (21X,1H*,9X,I1,12H. NO DIRECT,3A4,9HINCIDENT  DIAG0580
  1     7HWITHIN,25HRECEIVER'S FIELD OF VIEW.,13X,1H*  DIAG0590
  2     /,21X,1H*,88X,1H*)                          DIAG0600
  1400 FORMAT (21X,1H*,9X,I1,25H. ANGLE OF INCIDENCE FOR,  DIAG0610
  1     3A4,8HGREATERTHAN 80.0 DEGREES; NO,12X,  DIAG0620
  2     1H*/,21X,1H*,13X,26HCALCULATIONS WILL BE MADE  DIAG0630
  3     14HFOR ITARG = 0.;35X,1H*/,21X,1H*,88X,1H*)  DIAG0640
  1500 FORMAT (21X,1H*,9X,I1,1H.,1X,2A4,14HDOES NOT FACE  DIAG0650
                                         )  DIAG0660
                                         )  DIAG0670
                                         )  DIAG0680
                                         )  DIAG0690
                                         )  DIAG0700

```

```
      9H RECEIVER, 45X, 1H*, /, 21X, 1H*, 88X, 1H*)          DIAG0710
1600 FORMAT (21X, 1H*, 9X, I1, 1H, 3A4, 22H ILLUMINATES BACK SIDE   DIAG0720
      3H OF , 7H TARGET, 33X, 1H*, /, 21X, 1H*, 88X, 1H*)          DIAG0730
1700 FORMAT (21X, 1H*, 9X, I1, 30H, THETA LESS THAN 90 DEGREES    DIAG0740
      10H AND 1TARG 13H = 1; THETA = , F6.4, 19X, 1H*, /,        DIAG0750
      21X, 1H*, 88X, 1H*)          DIAG0760
1800 FORMAT (21X, 1H*, 9X, I1, 25H, THETA GREATER THAN 90          DIAG0770
      13H DEGREES AND 19H ITARG = 0; THETA = , F7.4,           DIAG0780
      13X, 1H*, /, 21X, 1H*, 88X, 1H*)          DIAG0790
1900 FORMAT (2(21X, 1H*, 88X, 1H*, /), 21X, 90(1H*))          DIAG0800
      RETURN          DIAG0810
      END          DIAG0820
```

```

SUBROUTINE INDAT(IEMISS, IHAZE, IM, LEN, ML)           IND00010
 1      MODEL, TBOUND, ISW, CLDHGT, BETAEX)             IND00020
COMMON /ALBED/A0(2), A1(2), IALB(2)                   IND00030
COMMON /M01/DUMMIE(715), MHOLD, NLHOLD, DUMMYS(1088), IND00040
+      TRANS(16,5), RADA(16,2), WAVE(16), SS(16),       IND00050
1      DIR(16), RADG(16), UTEM(16), UTRF(16), BK(16),   IND00060
2      PATHR(16), UERF(16), PATHR2(16), TOT(12), BKG(16)  IND00070
COMMON /BKDAT/ALT, THETA, PHI, SANG2, ZENTH, PHASE, ALB  IND00080
COMMON /COMI1/ISORC, ITARG, IWN, JHL                 IND00090
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK    IND00100
COMMON /CTARG/RTARG, COSX, COSY, COSZ                IND00110
COMMON /EMISS/EM(2), TM(2)                           IND00120
COMMON /IOUNIT/I0IN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU IND00130
COMMON /GEOMET/PTS(15), IGEOISW                     IND00140
COMMON /BASBOT/ANG(65), SUM(65), NWL(16), NWVL, ALBB(16), BS(16), IND00150
1 BE(16), SINGWY, PFC(65), LMAM
DIMENSION DAT(?), TYPE(?)
DATA TYPE /4HENVR, 4HEMIS, 4HATM, 4HTARG, 4HREFL, 4HSENS, 4HGO /
DATA IZERO /0/                                     IND00160
IND00170
IND00180
IND00190
IND00200
SUBROUTINE INDAT IS CALLED UPON TO A) READ INPUT CONTROL
PARAMETERS, WITH CARD ORDER INDEPENDENT INPUT (SEE SPOT
FOR MORE DETAIL) AND, B) TO READ VALUES OF THE PHASE
FUNCTION AT SPECIFIED ID AND WAVELENGTH.
NOTE. IF ISORC LT 2 OR ITARG LT 1 EM(1), TM(1), EM(2), TM(2)
ARE NOT NEEDED.
IF ITARG LT 1 A0(), A1(), IALB() ARE NOT NEEDED.          IND00210
IND00220
IND00230
IND00240
IND00250
IND00260
IND00270
IND00280
IND00290
IND00300
IND00310
IND00320
IND00330
IND00340
IND00350
IND00360
IND00370
IND00380
IND00390
IND00400
IND00410
IND00420
IND00430
IND00440
IND00450
IND00460
IND00470
IND00480
IND00490
IND00500
IND00510
IND00520
IND00530
IND00540
IND00550
IND00560
IND00570
IND00580
IND00590
IND00600
IND00610
IND00620
IND00630
IND00640
IND00650
IND00660
IND00670
IND00680
IND00690
IND00700
ISW=1 ON SECOND CALL TO INDAT
IF (ISW.EQ.1) GO TO 400
IF (IZERO.GT.0) GO TO 9
ISORC=0
ITARG=0
MODEL=0
IHAZE=0
NLAM=0
EM(1)=0.
TM(1)=0.
EM(2)=0.
TM(2)=0.
ZENTH=0.
CLDHGT=0.
PHASE=0.
RTARG=0.
COSX=0.
COSY=0.
COSZ=0.
A0(1)=0.
A1(1)=0.
IALB(1)=0
A0(2)=0.
A1(2)=0.
IALB(2)=0
ALT=0.
THETA=0.
PHI=0.
SANG2=0.
IZERO=1
CONTINUE
9  WRITE( IOOUT, 600)
 600 FORMAT(1H0, 'SPOT CONTROL CARDS READ FOR THIS RUN: ')
DO 10 I=1,?
READ (I0IN, 11) (DAT(J), J=1,7)
11  FORMAT (A4, 6X, 7E10.4)
WRITE( IOOUT, 610)(DAT(J), J=1,7)
610 FORMAT(1H, A4, 6X, 7E10.4)
IF (DAT(1).EQ. TYPE(1)) GO TO 1
IF (DAT(1).EQ. TYPE(2)) GO TO 2
IF (DAT(1).EQ. TYPE(3)) GO TO 3
IF (DAT(1).EQ. TYPE(4)) GO TO 4

```

```

IF <DAT(1)>.EQ.TYPE(5) GO TO 5          IND00710
IF <DAT(1)>.EQ.TYPE(6) GO TO 6          IND00720
IF <DAT(1)>.EQ.TYPE(?) GO TO 7          IND00730
C      ERROR RETURN                      IND00740
101   WRITE (IOUUT,101)                      IND00750
      101H INCORRECT INPUT CARD FOR SPOT, CONTROL RETURNED , IND00760
      1 18HTD MAIN FROM INDAT             IND00770
      IHAZE=9                           IND00780
      RETURN                           IND00790
C      OPERATING ENVIRONMENT           IND00800
      ISORC=IFIX(DAT(2))                IND00810
      ITARG=IFIX(DAT(3))                IND00820
      IHAZE=IFIX(DAT(4))                IND00830
      MODEL=IFIX(DAT(5))                IND00840
      NLAM=IFIX(DAT(6))                IND00850
      ID=IFIX(DAT(?))                 IND00860
      GO TO 8                           IND00870
C      EMISSIVITY AND TEMPERATURE OF GROUND AND TARGET, RESPECTIVELY IND00880
      2      EM(1)=DAT(2)                  IND00890
      TM(1)=DAT(3)                    IND00900
      EM(2)=DAT(4)                  IND00910
      TM(2)=DAT(5)                  IND00920
      TBOUND=TM(1)                  IND00930
      GO TO 8                           IND00940
C      INCIDENT ZENITH ANGLE OF RADIATION, CLOUD BOTTOM HEIGHT,        IND00950
      PHASE ANGLE OF MOON, OPTIONAL EXTN COEF (VALID WHEN IHAZE=8)  IND00960
      3      ZENTH=DAT(2)                IND00970
      CLDHGT=DAT(3)                  IND00980
      PHASE=DAT(4)                  IND00990
      BETAEX=DAT(5)                  IND01000
      GO TO 8                           IND01010
C      TARGET PROPERTIES              IND01020
      4      RTARG=DAT(2)                IND01030
      COSX=DAT(3)                  IND01040
      COSY=DAT(4)                  IND01050
      COSZ=DAT(5)                  IND01060
      GO TO 8                           IND01070
C      ALBEDO COEFFICIENTS AND TYPE OF REFLECTION SURFACE FOR       IND01080
      GROUND AND TARGET, RESPECTIVELY           IND01090
      5      A0(1)=DAT(2)                IND01100
      A1(1)=DAT(3)                  IND01110
      IALB(1)=IFIX(DAT(4))            IND01120
      A0(2)=DAT(5)                  IND01130
      A1(2)=DAT(6)                  IND01140
      IALB(2)=IFIX(DAT(7))            IND01150
      GO TO 8                           IND01160
C      SENSOR CHARACTERISTICS         IND01170
      6      ALT=DAT(2)                  IND01180
      THETA=DAT(3)                  IND01190
      PHI=DAT(4)                   IND01200
      GO TO 8                           IND01210
CCCCC EXPECTING INPUT AZIMUTH IN METEOROLOGICAL CONVENTION          IND01220
      (I.E., N = 0 DEG, E = 90 DEG, S = 180 DEG, W = 270 DEG),      IND01230
      SO CONVERT TO MATHEMATICAL CONVENTION FOR PURPOSES OF          IND01240
      SPOT (ASSUMING Y-AXIS IS POSITIVE NORTHWARD, X-AXIS POSI-      IND01250
      TIVE EASTWARD).          IND01260
      IND01270
      PHI=90.-PHI                  IND01280
      SANGZ=DAT(5)                  IND01290
      CONTINUE                         IND01300
      CONTINUE                         IND01310
      CONTINUE                         IND01320
      IEMISS=0                         IND01330
      IM=0                            IND01340
      LEN=0                           IND01350
      ML=0                            IND01360
      IF (<ISORC.GT.1>) IEMISS=1          IND01370
      COSX=COS(COSX*PIRAD)            IND01380
      COSY=COS(COSY*PIRAD)            IND01390
      COSZ=COS(COSZ*PIRAD)            IND01400

```

```

C GEOMETRICAL OPTION
IF(IGEOSW.NE.1)GO TO 311
RTARG=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+
+(PTS(6)-PTS(3))**2)
THETA=ACOS((PTS(3)-PTS(6))/RTARG)
RTDCON=1.0/PIRAD
THETA=THETA*RTDCON
ALT=PTS(6)
DELX=PTS(1)-PTS(4)
DELY=PTS(2)-PTS(5)
HDIS=SQRT(DELX**2+DELY**2)
PHI=ACOS(DELX/HDIS)
PHI=PHI*RTDCON
IF(DELY.LT.0.0)PHI=360.0-PHI
IND01410
IND01420
IND01430
IND01440
IND01450
IND01460
IND01470
IND01480
IND01490
IND01500
IND01510
IND01520
IND01530
IND01540
IND01550
IND01560
IND01570
IND01580
IND01590
IND01600
IND01610
IND01620
IND01630
IND01640
IND01650
311 CONTINUE
IF (ITARG.EQ.1) RTARG=ALT/ABS(COS(THETA*PIRAD))
IF (ITARG.EQ.0) RTARG=1000.0
RETURN
400 CONTINUE
REWIND IPHFUN
DO 500 I=1,IWN
PF(I)=0
IF (NLAM.NE.0) CALL PFUNC(ID)
RETURN
END

```

```

SUBROUTINE OUTPUT(MODEL,IHAZE,CLDHGT)
LOGICAL LMNLT
COMMON /M01/ DUMMIE(715),MHOLD,NLHOLD,DUMMYS(1088),
+           TRANS(16,5),RADAC(16,2),WAVE(16),SS(16),
1           DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),
2           PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)
COMMON /BKDAT/ ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
COMMON /COM1/ ISORC,ITARG,IWN,JHL
COMMON /ANSW2/TTR(16),TBR(16),CNTRST(16)
COMMON /HOLRTH/          HITARG(8,3),HISORC(6,5),
1           HMODEL(5,6),HIGHT(3),HMNLT(3),HSNLT(3),
2           HTRGT(2),HTYPE(2),HGRND(2)
COMMON /IOUNIT/ IOIN,IOOUT,IPHFUN,LOUNIT,NDIR TU,NCLIM T,KSTOR,NPLOT U
DATA HGRND/4H GRO,4HUND/
DATA HISORC/4HSUNL,4HIGHT,4H ONL,4HY   ,4H   ,4H
1   4HMOON,4HLIGH,4HT ON,4HLY   ,4H   ,4H   ,4HEMIS/
2   4HSION,4H ONL,4HY   ,4H   ,4H   ,4HSUNL,4HIGHT,
3   4H AND,4H EMI,4HSSIO,4HN   ,4HMOON,4HLIGH,4HT AN,
4   4HD EM,4HISSI,4HON/
DATA HITARG/4HNO R,4HEFLE,4HCTAN,4HCE,4H
1   4H   ,4HGROU,4HND R,4HEFLE,4HCTAN,4HCE ,
2   4H EMI,4HSSIO,4HN   ,4HTARG,4HET R,4HEFLE,4HCTAN,
3   4HCE ,4H EMI,4HSSIO,4HN   /
DATA HMODEL/4HTROP,4HICAL,4H   ,4H   ,4H   ,4HMIDA,
1   4HLTIT,4HDE  ,4HSUMM,4HER   ,4HMIDA,4HLTIT,4HDE ,
2   4HWINT,4HER   ,4HSUBA,4HRCTI,4HC SU,4HMMER,4H
3   4HSUBA,4HRCTI,4HC WI,4HNTER,4H   ,4H1962,4H U.S,
4   4H ST,4HANDA,4HRD /
DATA HMNLT/4H MO,4HONLI,4HGHT /
DATA HSNLT/4H SU,4HNLIG,4HHT /
DATA HTRGT/4H TAR,4HGET /
DATA HTYPE/4H   ,4H   /
DATA LMNLT/.FALSE./
IF (ISORC.NE.0.AND.ISORC.NE.3) GO TO 200
DO 100 I=1,3
100 HLIGHT(I)=HSNLT(I)
GO TO 400
200 DO 300 I=1,3
300 HLIGHT(I)=HMNLT(I)
LMNLT=.TRUE.
400 IF (ITARG.EQ.0) GO TO 600
IF (ITARG.EQ.2) GO TO 500
HTYPE(1)=HGRND(1)
HTYPE(2)=HGRND(2)
GO TO 600
500 HTYPE(1)=HTRGT(1)
HTYPE(2)=HTRGT(2)
600 CALL DIAG
IF (MODEL.GT.7) GO TO 700
WRITE (IOOUT,1400) ISORC,(HISORC(I,ISORC+1),I=1,6),
1 ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,
2 (HMODEL(I,MODEL),I=1,5),IHAZE
GO TO 750
700 IF(MODEL.EQ.8) WRITE (IOOUT,1450) ISORC,(HISORC(I,ISORC+1),I=1,6),
1 ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE
IF(MODEL.EQ.9) WRITE (IOOUT,1500) ISORC,(HISORC(I,ISORC+1),I=1,6),
1 ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE
750 IF (IHAZE.GT.0) WRITE (IOOUT,1600) CLDHGT
800 WRITE (IOOUT,1700)
IF (LMNLT) WRITE (IOOUT,1800) PHASE
WRITE (IOOUT,1900) HLIGHT
DO 900 I=1,IWN
NW=10000./WAVE(I)
900 WRITE (IOOUT,2000) WAVE(I),NW,SS(I),BK(I),BKG(I)
WRITE (IOOUT,2100)
DO 1000 I=1,IWN
NW=10000./WAVE(I)
1000 WRITE (IOOUT,2200) WAVE(I),NW,UTEM(I),UTRF(I),
1 RADAC(I,2),PATHR2(I),TTR(I)
WRITE (IOOUT,2300)

```

```

DO 1100 I=1,IWN
NW=10000./WAVE(I)
1100 WRITE (IOOUT,2200) WAVE(I),NW,RADG(I),UERF(I),
      RADAC(I,1),PATHR(I),TBR(I)
      WRITE (IOOUT,2400) HLIGHT
      IF (LMNLT) WRITE (IOOUT,1800) PHASE
      WRITE (IOOUT,2500) HLIGHT,HLIGHT
      DO 1200 I=1,IWN
      NW=10000./WAVE(I)
1200 WRITE (IOOUT,2600) WAVE(I),NW,SSC(I),DIR(I)
      WRITE (IOOUT,2700)
      DO 1300 I=1,IWN
      NW=10000./WAVE(I)
1300 WRITE (IOOUT,2800) WAVE(I),NW,TTR(I),TBR(I),CNTRST(I)
      WRITE (IOOUT,2900) (TOT(I),I=1,10),TOT(12),HLIGHT,
      TOT(11)
      RETURN

1400 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1      8HFOLLOWS:,/,43X,10(1H-),6H -- ,7(1H-),2X,
2      10(1H-),2X,7(1H-),//,43X,9HPARAMETER,3X,
3      5HVALUE,3X,11HDESCRIPTION/,43X,9(1H-),3X,5(
4      1H-),3X,11(1H-),//,45X,5HSOURCE,7X,I1,5X,6A4,//
5      ,45X,5HITARG,7X,I1,5X,8A4,/,45X,5HMODEL,7X,
6      ,11,5X,5A4,/,45X,5HIAZES,7X,11,/)
1450 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1      8HFOLLOWS:,/,43X,10(1H-),6H -- ,7(1H-),2X,
2      10(1H-),2X,7(1H-),//,43X,9HPARAMETER,3X,
3      5HVALUE,3X,11HDESCRIPTION/,43X,9(1H-),3X,5(
4      1H-),3X,11(1H-),//,45X,5HSOURCE,7X,I1,5X,6A4,//
5      ,45X,5HITARG,7X,I1,5X,8A4,/,45X,5HMODEL,7X,
6      ,11,5X,32HSISRAELI STANDARD (YEAR, DAYTIME)/,45X,5HIAZES,7X,I1/)
1500 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1      8HFOLLOWS:,/,43X,10(1H-),6H -- ,7(1H-),2X,
2      10(1H-),2X,7(1H-),//,43X,9HPARAMETER,3X,
3      5HVALUE,3X,11HDESCRIPTION/,43X,9(1H-),3X,5(
4      1H-),3X,11(1H-),//,45X,5HSOURCE,7X,I1,5X,6A4,//
5      ,45X,5HITARG,7X,I1,5X,8A4,/,45X,5HMODEL,7X,
6      ,11,5X,34HSISRAELI STANDARD (YEAR, NIGHTTIME)/,45X,5HIAZES,7X,I1/)
1600 FORMAT (1H ,45X,22HCLOUD BOTTOM HEIGHT = ,F5.3,3H KM)
1700 FORMAT (1H1,56X,19HSOURCE INTENSITIES/,56X,7H-----
1      12H -----//)
1800 FORMAT (44X,27HPHASE ANGLE FOR MOONLIGHT: ,F6.2,
1      10H (DEGREES),//)
1900 FORMAT (15X,10HWAVENUMBER,3X,10HWAVENUMBER,3X,3A4,
1      6HSOURCE,11X,13HTARGET SOURCE,15X,7HGROUND
2      6HSOURCE,/,15X,9H MICRONS),6X,6H(CM-1),11X,
3      8HSTRENGTH,17X,8HSTRENGTH,20X,8HSTRENGTH/
4      ,41X,20H(WATTS M-2 MICRON-1),2(3X,9H(WATTS M-
5      16H2 MICRON-1 SR-1)),/,15X,10(1H-),3X,10(1H-),
6      3X,20(1H-),3X,25(1H-),3X,25(1H-),//)
2000 FORMAT (15X,1PE10.4,3X,17,11X,1PE10.4,15X,1PE10.4,1BX,
1      1PE10.4)
2100 FORMAT (1H1,46X,33HCOMPONENTS FOR RADIANCE FROM
1      6HTARGET/,46X,29H----- --
2      10H-----//,53X,23H(WATTS M-2 MICRON-1 SR-
3      2H1),//,22X,10HWAVENUMBER,3X,10HWAVENUMBER,
4      5X,6HTARGET,7X,6HTARGET,7X,7HPARTIAL,6X,
5      7HPARTIAL,6X,5HTOTAL,/,22X,9H(MICRONS),6X,
6      6H(CM-1),6X,BHEMISSION,4X,11HREFLECTANCE,2X,
7      11HATMOSPHERIC,5X,4HPATH,8X,6HTARGET,/,75X,
8      BHEMISSION,5X,8HRADIANCE,5X,8HRADIANCE/
9      ,22X,10(1H-),3X,10(1H-),4X,8(1H-),4X,11(1H-),
X      ,2X,11(1H-),3X,8(1H-),5X,8(1H-),//)
2200 FORMAT (22X,1PE10.4,3X,17,3X,1P5E13.4)
2300 FORMAT (1H1,47X,29HCOMPONENTS FOR BACKGROUND
1      8HRADIANCE/,47X,27H----- --
2      10H-----//,53X,23H(WATTS M-2 MICRON-1 SR-
3      2H1),//,22X,10HWAVENUMBER,3X,10HWAVENUMBER,
4      5X,6HGROUND,7X,6HGROUND,8X,5HTOTAL,8X,5HTOTAL,

```

```

5      7X,5HTOTAL,/,,22X,9H(MICRONS),6X,6H(CM-1),6X,          OUTP1410
6      8HEMISSION,4X,11HREFLECTANCE,2X,11HATMOSPHERIC        OUTP1420
7      ,5X,4HPATH,6X,10HBACKGROUND,/,,75X,8HEMISSION,        OUTP1430
8      2(5X,8HRADIANCE),/,,22X,10(1H-),3X,10(1H-),4X,        OUTP1440
9      8(1H-),4X,11(1H-),2X,11(1H-),3X,8(1H-),4X,10(        OUTP1450
X      1H-),//)                                              OUTP1460
2400 FORMAT <1H1,58X,6HDIRECT,3A4,/,,57X,6(1H-),1X,12(1H-),/        OUTP1470
1      /,,56X,20H(WATTS M-2 MICRON-1),//>                  OUTP1480
2500 FORMAT <41X,10HWAVELENGTH,3X,10HWAVENUMBER,2X,3A4,1X,        OUTP1490
1      3A4,/,,41X,9H(MICRONS),6X,6H(CM-1),2X,6HSOURCE,        OUTP1500
2      8X,4HFLUX,/,,68X,8HSTRENGTH,/,,41X,10(1H-),3X,        OUTP1510
3      10(1H-),2X,12(1H-),1X,12(1H-),//>                  OUTP1520
2600 FORMAT <41X,1PE10.4,3X,I7,6X,1PE10.4,3X,1PE10.4>        OUTP1530
2700 FORMAT <1H1,58X,15HTOTAL RADIANCE,/,,58X,11H-----        OUTP1540
1      4H----//,,53X,25H(WATTS M-2 MICRON-1 SR-1),///        OUTP1550
2      ,35X,10HWAVELENGTH,3X,10HWAVENUMBER,5X,                OUTP1560
3      6HTARGET,5X,10HBACKGROUND,4X,8HCONTRAST,/,,            OUTP1570
4      ,35X,9H(MICRONS),6X,6H(CM-1),33X,5HRATIO,/,,        OUTP1580
5      ,35X,10(1H-),3X,10(1H-),5X,6(1H-),5X,10(1H-),        OUTP1590
6      4X,8(1H-),//)                                         OUTP1600
2800 FORMAT <35X,1PE10.4,3X,I7,3X,1P3E13.4>                 OUTP1610
2900 FORMAT <1H1,46X,30HDETECTOR-RESPONSE WAVELENGTH-        OUTP1620
1      10HINTEGRATED,/,,46X,17(1H-),2X,21(1H-),//,          OUTP1630
2      ,58X,16H(WATTS M-2 SR-1),//,,46X,7HTARGET           OUTP1640
3      8HEMISSION,16X,1PE10.4,/,,46X,7HTARGET             OUTP1650
4      11HREFLECTANCE,13X,1PE10.4,/,,46X,8HPARTIAL        OUTP1660
5      20HATMOSPHERIC EMISSION,3X,1PE10.4,/,,46X,           OUTP1670
6      21HPARTIAL PATH RADIANCE,10X,1PE10.4,/,,46X,          OUTP1680
7      21HTOTAL TARGET RADIANCE,10X,1PE10.4,/,,46X,          OUTP1690
8      15HGROUND EMISSION,16X,1PE10.4,/,,46X,7HGROUND       OUTP1700
9      11HREFLECTANCE,13X,1PE10.4,/,,46X,6HTOTAL          OUTP1710
X      20HATMOSPHERIC EMISSION,5X,1PE10.4,/,,46X,           OUTP1720
1      19HTOTAL PATH RADIANCE,12X,1PE10.4,/,,46X,           OUTP1730
2      25HTOTAL BACKGROUND RADIANCE,6X,1PE10.4,/,,           OUTP1740
3      ,46X,41(1H*),//,,46X,8HCONTRAST,22X,1PE11.4,/,,     OUTP1750
4      ,46X,41(1H*),//,,46X,6HDIRECT,3A4,13X,1PE10.4,/,,   OUTP1760
5      ,46X,11H(WATTS M-2)>                                OUTP1770
END                                                 OUTP1780

```

```

SUBROUTINE PATHRD(CTHED,HP,RT,IO,IHAZE,NR,
1 IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV) PAD00010
LOGICAL HORIZ PAD00020
DIMENSION TR1(16),TR2(16),DUMMY(16),ANS(16) PAD00030
COMMON /BASBOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),
1 BE(16),SINGWV,SAER(65),LOUM PAD00040
COMMON /BKDAT/ ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB PAD00050
COMMON /CGEOM/COSGM,COSBT,CSA PAD00060
COMMON /COMI1/ISORC,ITARG,IWN,JHL PAD00070
COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,
+ RRS(16,34),SCOE(16,34), PAD00080
+ TRANS(16,5),RADAC(16,2),WAVE(16),SS(16), PAD00090
1 DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16), PAD00100
2 PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16) PAD00110
C ***** CONST1 = (1.-.0295)*(5*3.)/(8.*PI)/(1.+5*.0295) PAD00120
C ***** CONST2 = .0295/(1.+5*.0295)*3./(8.*PI) PAD00130
DATA CONST1,CONST2/.0570805145,.0034701189/ PAD00140
NLL=NL-1 PAD00150
SRAYL=CONST1*(1.+CSA*CSA)+CONST2 PAD00160
C ***** INITIALIZE VARIABLES *****
DO 800 IW=1,IWN PAD00170
800 PATHR(IW)=0.0 PAD00180
DO 900 J=2,NLL PAD00190
IF (HP.LT.Z(J)) GO TO 1000 PAD00200
900 CONTINUE PAD00210
J=NLL PAD00220
1000 JU=J PAD00230
JL=JU-1 PAD00240
HORIZ=.FALSE. PAD00250
DS1=0.0 PAD00260
H2=HP PAD00270
RAT=(H2-Z(JL))/(Z(JU)-Z(JL)) PAD00280
GO TO 1100,1200,1300,IO PAD00290
1100 IX=RT*1.99999+1 PAD00300
DX=RT/FLOAT(IX) PAD00310
DH=DX*CTHED PAD00320
HORIZ=ABS(CTHED).LT.1.E-3 PAD00330
DIST=0.5*DX PAD00340
GO TO 1400 PAD00350
1200 IX=NLL-JL PAD00360
DH=Z(JU)-HP PAD00370
DX=DH/CTHED PAD00380
GO TO 1400 PAD00390
C HORIZONTAL PATH FOLLOWS
1300 IX=100 PAD00400
DIST=DIST-0.5*DX+0.25 PAD00410
DX=0.5 PAD00420
HORIZ=.TRUE. PAD00430
GO TO 1500 PAD00440
1400 H2=HP+DH*0.5 PAD00450
1500 DO 2600 K=1,IX PAD00460
IF (HORIZ) GO TO 2000 PAD00470
CALL LT4M(HP,H2,THETA,2,2,TR1,DUMMY,DUMMY,
1 IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,ICLMAT,IERR, PAD00480
2 NR,IHAZE,MULDV) PAD00490
CALL LT4M(H2,DUM,ZENTH,3,2,TR2,DUMMY,DUMMY,
1 IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,ICLMAT,IERR, PAD00500
2 NR,IHAZE,MULDV) PAD00510
IF (IO.EQ.2) GO TO 1800 PAD00520
DO 1600 J=2,NLL PAD00530
IF (H2.LT.Z(J)) GO TO 1700 PAD00540
1600 CONTINUE PAD00550
J=NLL PAD00560
1700 JU=J PAD00570
JL=JU-1 PAD00580
RAT=(H2-Z(JL))/(Z(JU)-Z(JL)) PAD00590
1800 DO 1900 IW=1,IWN PAD00600
SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL)) PAD00610
RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL)) PAD00620
SSCAT=RS*SRAYL+(1.0-RS)*SAER(IW) PAD00630
PAD00640
PAD00650
PAD00660
PAD00670
PAD00680
PAD00690
PAD00700

```

```

1900 PATHR(IW)=PATHR(IW)+TR1(IW)*TR2(IW)*SC*SSCAT*DX          PAD00710
      GO TO 2400,2500,2300),IO
2000 CONTINUE
      CALL LT4M(HP,DUM,DIST,1,2,TR1,DUMMY,DUMMY,
      1 IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,ICLMAT,IERR,
      2 NR,IHAZE,MULDV)
      IF (K.EQ.1) CALL LT4M(HP,DUM,ZENTH,3,2,TR2,DUMMY,DUMMY,
      1 IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,
      2 ICLMAT,IERR,
      2 NR,IHAZE,MULDV)
      DS=0.0
      DO 2200 IW=1,IWN
      IF (K.GT.1) GO TO 2100
      SC=S00E(IW,JL)+RAT*(S00E(IW,JU)-S00E(IW,JL))
      RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))
      ANS(IW)=TR2(IW)*SC*(RS*SRAYL+(1.0-RS)*SAER(IW))
2100 DPATH=ANS(IW)*TR1(IW)*DX
      DS=DS+DPATH
2200 PATHR(IW)=PATHR(IW)+DPATH
      DS1=DS1+DS
      DS=DS*0.5/DX
      IF (IO.EQ.3,AND,K.GT.1,AND,DS/DS1.LT.0.001) RETURN
      IK=(K/20)*20
      IF (IO.EQ.3,AND,IK.EQ.K) DX=DX*2.0
2300 DIST=DIST+DX
      GO TO 2600
2400 H2=H2+DH
      GO TO 2600
2500 JU=JU+1
      JL=JL+1
      IF (K.EQ.1) RAT=0.5
      H2=(Z(JU)+Z(JL))*0.5
      DX=(Z(JU)-Z(JL))/CTHED
2600 CONTINUE
      RETURN
      END

```

```
SUBROUTINE ZERO
COMMON /M01/ DUMMIE(715),M,NL,DUMMYS(1088),RDATA(300)
DO 100 I=1,300
RDATA(I)=0.0
100 CONTINUE
RETURN
END
```

```
ZER00010
ZER00020
ZER00030
ZER00040
ZER00050
ZER00060
ZER00070
```

SUBROUTINE CLIMAT(LOCAT,MONTH,NHOUR,IWIND,NPRT,TEMP,PRESS,RH,AH,DPCLI00010
1,VIS,WNDVEL,WINDIR,IPASCT) CLI00020

***** CLI00030

CLIMATOLOGY MODULE - CLIMAT CLI00040

PURPOSE - TO PROVIDE THE CENTRAL EUROPEAN AND MID-EASTERN CLI00050

CLIMATOLOGY DATA REQUIRED BY OTHER MODULES OF EOSAEL. CLI00060

PARAMETER DESCRIPTION CLI00070

LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER CLI00080

(1-4) FOR CENTRAL EUROPE AND CLI00090

(5-10) FOR MID-EAST. CLI00100

REGION 1 - EUROPEAN LOWLANDS, CLI00110

REGION 2 - EUROPEAN RHINE VALLEY, CLI00120

REGION 3 - EUROPEAN HIGHLANDS, CLI00130

REGION 4 - EUROPEAN ALPINE, CLI00140

REGION 5 - MIDEAST DESERTS, CLI00150

REGION 6 - MIDEAST COASTAL, CLI00160

REGION 7 - MIDEAST PERSIAN GULF, CLI00170

REGION 8 - MIDEAST RED SEA, CLI00180

REGION 9 - MIDEAST EASTERN MOUNTAINS, AND CLI00190

REGION 10 - MIDEAST INDUS VALLEY. CLI00200

MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR. CLI00210

MONTH IS USED TO SELECT THE SEASON WHICH IS CLI00220

APPLICABLE TO THE REGION LOCAT. CLI00230

NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL CLI00240

STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OF CLI00250

FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14, CLI00260

AND 15-19. CLI00270

IWIND - *** NOT USED *** CLI00280

NPRT - A PRINT SELECTOR. CLI00290

NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA, CLI00300

NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD CLI00310

DEVIATIONS, AND PERCENT OCCURRENCES. CLI00320

TEMP - MEAN TEMPERATURE (C). CLI00330

PRESS - MEAN SEA LEVEL PRESSURE (MB). CLI00340

RH - MEAN RELATIVE HUMIDITY (PERCENT). CLI00350

AH - MEAN ABSOLUTE HUMIDITY (GM/CU M). CLI00360

DP - MEAN DEW-POINT TEMPERATURE (C). CLI00370

VIS - MEAN HORIZONTAL VISIBILITY (KM). CLI00380

WNDVEL - MEAN WIND SPEED (MPS). CLI00390

WINDIR - MOST PROBABLE WIND DIRECTION (DEGREES). WINDIR IS CLI00400

GIVEN IN 30 DEGREE INCREMENTS (015,045,075,...,345). CLI00410

IPASCT - INDICATOR (1-6) FOR THE MOST PROBABLE PASQUILL CLI00420

STABILITY CATEGORY (A-F). CLI00430

CLDHT - MEAN CLOUD HEIGHT (KM). CLI00440

CLDCVR - MEAN TOTAL CLOUD COVER (PERCENT). CLI00450

WNDDIR - WIND DIRECTION (DEGREES). CLI00460

SUBROUTINES AND FUNCTIONS - NONE CLI00470

CARD INPUT - NONE CLI00480

TAPE INPUT - YES. BE SURE TO ASSIGN THE CLIMATOLOGY DATA TAPE TO CLI00490
UNIT NCLIMT. CLI00500

COMMON /IOUNIT/I0IN, I0OUT, IPHFUN, LOUNIT, NDINTU, NCLIMT, KSTOR, NPLOTU, CLI00510
DIMENSION REGION(70), SEASON(8), HOUR(8), DATA(18), DIR(13) CLI00520

DATA REGION/4HEURO, 4HPEAN, 4H LOW, 4HLAND, 4HS CLI00530

1 2*4H , 4HEURO, 4HPEAN, 4H RHI, 4HNE Y, 4HALLE, 4HY , 4H , 4HEURO, CLI00540

2 4HPEAN, 4H HIG, 4HHLAN, 4HDS , 2*4H , 4HEURO, 4HPEAN, 4H ALP, 4HNE , 4H , 4HEURO, CLI00550

3 3*4H , 4HMIDE, 4HAST , 4HDESE, 4HRTS , 3*4H , 4HMIDE, 4HAST , 4HCOACLI00560

1 2*4H , 4HEURO, 4HPEAN, 4H RHI, 4HNE Y, 4HALLE, 4HY , 4H , 4HEURO, CLI00570

2 4HPEAN, 4H HIG, 4HHLAN, 4HDS , 2*4H , 4HEURO, 4HPEAN, 4H ALP, 4HNE , 4H , 4HEURO, CLI00580

3 3*4H , 4HMIDE, 4HAST , 4HDESE, 4HRTS , 3*4H , 4HMIDE, 4HAST , 4HCOACLI00590

```

4S,4HTAL,3*4H ,4HMIDE,4HAST ,4HPERS,4HIAN ,4HGULF,2*4H ,4HMICLI00710
5DE,4HAST ,4HRED ,4HSEA ,3*4H ,4HMIDE,4HAST ,4HEAST,4HERN ,4HMOUCCLI00720
6N,4HTAIN,4HS ,4HMIDE,4HAST ,4HINDU,4HS VA,4HLLEY,2*4H / CLI00730
DATA SEASON/4HWINT,4HER ,4HSPRI,4HNG ,4HSUMM,4HER ,4HAUTU,4HMM CLI00740
1 /
DATA HOUR/4H20-0,4H2 ,4H03-0,4H9 ,4H10-1,4H4 ,4H15-1,4H9 /CLI00760
DATA DIR/4H 015,4H 045,4H 075,4H 105,4H 135,4H 165,4H 195,4H 225, CLI00770
14H 255,4H 285,4H 315,4H 345,4H VBL/ CLI00780
CLI00790
CLI00800
CLI00810
CLI00820
CLI00830
CLI00840
CLI00850
CLI00860
CLI00870
CLI00880
CLI00890
CLI00900
CLI00910
CLI00920
CLI00930
CLI00940
CLI00950
CLI00960
CLI00970
CLI00980
CLI00990
CLI01000
CLI01010
CLI01020
CLI01030
CLI01040
CLI01050
CLI01060
CLI01070
CLI01080
CLI01090
CLI01100
CLI01110
CLI01120
CLI01130
CLI01140
CLI01150
CLI01160
CLI01170
CLI01180
CLI01190
CLI01200
CLI01210
CLI01220
CLI01230
CLI01240
CLI01250
CLI01260
CLI01270
CLI01280
CLI01290
CLI01300
CLI01310
CLI01320
CLI01330
CLI01340
CLI01350
CLI01360
CLI01370
CLI01380
CLI01390
CLI01400

POSITION THE TAPE NCLIMT FOR READING
REWIND NCLIMT
SKIP OVER ALL DATA FOR REGIONS 1,2,...,LOCAT-1
IF(LOCAT.LT.1,OR.LOCAT.GT.10) LOCAT=1
LSKIP=1056*(LOCAT-1)

SKIP OVER DATA FOR SEASONS 1,2,...,SEASON-1 FOR REGION LOCAT
NSEASN=1
IF(MONTH.GE.3.AND.MONTH.LE.5) NSEASN=2
IF(MONTH.GE.6.AND.MONTH.LE.8) NSEASN=3
IF(MONTH.GE.9.AND.MONTH.LE.10) NSEASN=4
IF(LOCAT.GE.5.AND.MONTH.EQ.11) NSEASN=4
NSKIP=LSKIP+176*(NSEASN-1)

SKIP OVER DATA FOR TIME PERIODS 0,1,...,PERIOD-1 FOR REGION
LOCAT DURING SEASON.
NTIME=1
IF(NHOUR.GE.3.AND.NHOUR.LE.9) NTIME=2
IF(NHOUR.GE.10.AND.NHOUR.LE.14) NTIME=3
IF(NHOUR.GE.15.AND.NHOUR.LE.19) NTIME=4
NSKIP=NSKIP+44*(NTIME-1)
IF(NSKIP.LE.0) GO TO 2
DO 1 J=1,NSKIP
READ(NCLIMT,9) A
1 CONTINUE

IF NPRT GT 0, PRINT A HEADING FOR THE THERMODYNAMIC DATA
2 CONTINUE
IF(NPRT.LE.0) GO TO 3
WRITE(I00UT,15)
ILOC=7*LOCAT-6
ILOC6=ILOC+6
WRITE(I00UT,10) (REGION(J),J=ILOC,ILOC6),SEASON(2*NSEASN-1),
1SEASON(2*NSEASN),HOUR(2*NTIME-1),HOUR(2*NTIME)

READ THE THERMODYNAMIC DATA FOR REGION LOCAT AT NTIME
DURING NSEASN.
3 DO 4 J=1,22
READ(NCLIMT,11) NCLASS,(DATA(K),K=1,18)
CONVERT FROM METERS TO KILOMETERS
DATA(6)=0.001*DATA(6)
DATA(10)=0.001*DATA(10)

IF NPRT GT 0, PRINT THE THERMODYNAMIC DATA
IF(NPRT.LE.0) GO TO 4
WRITE(I00UT,12) NCLASS,(DATA(K),K=1,18)
4 CONTINUE

EXTRACT THE VALUES OF TEMP, PRESS, RH, AH, DP, VIS, AND
WNDVEL.

```

```

C TEMP=DATA(2)
C DP=DATA(3)
C AH=DATA(4)
C RH=DATA(5)
C VIS=DATA(6)
C PRESS=DATA(7)
C WNDVEL=DATA(8)

      DETERMINE THE VALUE OF IPASCT

IPASCT=1
FREQ=DATA(13)
DO 5 J=2,6
IF(DATA(J+12),LE,FREQ) GO TO 5
FREQ=DATA(J+12)
IPASCT=J
5 CONTINUE

      GET TO THE WIND DATA ON NCLIMT

NSKIP=(NSKIP-LSKIP)/44
NSKIP=44*(15-NSKIP)+22*NSKIP
DO 16 J=1,NSKIP
READ(NCLIMT,9) A
16 CONTINUE

      IF NPRT GT 0, PRINT A HEADING FOR THE WIND DATA

IF(NPRT,LE,0) GO TO 6
WRITE(I00UT,13) (DIR(J),J=1,13)

      READ THE WIND DATA FOR REGION LOCAT AT NTIME DURING NSEASN

6 DO 7 J=1,22
READ(NCLIMT,14) NCCLASS,(DATA(K),K=1,14)

      IF NPRT GT 0, PRINT THE WIND DATA

IF(NPRT,LE,0) GO TO 7
WRITE(I00UT,17) NCCLASS,(DATA(K),K=1,14)
7 CONTINUE

      DETERMINE THE VALUE OF WINDIR

NDIR=1
FREQ=DATA(2)
DO 8 J=2,12
IF(DATA(J+1),LE,FREQ) GO TO 8
FREQ=DATA(J+1)
NDIR=J
8 CONTINUE
WINDIR=30*NDIR-15

      RETURN FROM CLIMAT

RETURN

      FORMAT STATEMENTS

9 FORMAT(A1)
10 FORMAT(25H1 EOSAEL CLIMATOLOGY FOR ,7A4,8H DURING ,2A4,4H AT ,2A4,
12H(LST), //126H CLASS FREQCY MEAN MEAN MEAN MEAN MEAN MEAN MEAN
2 MEAN MEAN/STDEV MEAN MEAN/STDEV FREQCY FREQCY FREQCY FREQCY FREQCY
30Y FREQCY/126H NO. CLASS TEMP DP AH RH VIS PCL
4RESS WNDVEL CLDHT CLDCVR A B C D E
5 F /126H (%) (C) (C) (GM/CU,M) (%) (KM) (%) (%)
6MB) (MPS) (KM) (%) (%) (%) (%) (%) (%) (%)
7 (%) /)
11 FORMAT(6X,I3,5F5.1,F7.0,F6.1,2F5.1,F6.0/8F5.1)
12 FORMAT(15,F9.1,4F7.1,F7.3,F7.1,F4.1,1H/,F4.1,F7.3,F6.1,1H/,F5.1,6F

```

17,1)
13 FORMAT(1H0/7H CLASS ,14(7H FREQCY)/14H NO. , CLASS,13(7H WNDDIR)CLI02110
1/10X,3H(%),1X,13(2X,A4,1X)/14X,13(3X,3H(%),1X)/) CLI02120
14 FORMAT(6X,I3,14F5.1) CLI02130
15 FORMAT(18H1CLIMATOLOGY MODEL///39H DEFINITIONS OF METEOROLOGICAL CLI02140
1CLASSES//48H CLASS 1 = FOG, HAZE AND MIST WITH VIS LT 1 KM./53H CCLI02150
2CLASS 2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM./53H CLASS 3 =CLI02160
3 FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM./48H CLASS 4 = FOG, HAZCLI02170
4E AND MIST WITH VIS GE 7 KM./34H CLASS 5 = DUST WITH VIS LT 3 KM.CLI02190
5/34H CLASS 6 = DUST WITH VIS GE 3 KM./53H CLASS 7 = DRIZZLE, RAICLI02200
6N AND TSTMWS WITH VIS LT 1 KM./58H CLASS 8 = DRIZZLE, RAIN AND TSTMWS WCLI02210
7MS WITH 1 LE VIS LT 3 KM./58H CLASS 9 = DRIZZLE, RAIN AND TSTMWS WITH CLI02220
8ITH 3 LE VIS LT 7 KM./53H CLASS 10 = DRIZZLE, RAIN AND TSTMWS WITH CLI02230
9VIS GE 7 KM./34H CLASS 11 = SNOW WITH VIS LT 1 KM./39H CLASS 12 = CLI02240
ASNOW WITH 1 LE VIS LT 3 KM./39H CLASS 13 = SNOW WITH 3 LE VIS LT 7CLI02250
8 KM./34H CLASS 14 = SNOW WITH VIS GE 7 KM./59H CLASS 15 = NO WEATHCLI02260
CER AND ABSOLUTE HUMIDITY LT 10 GM/CU M./59H CLASS 16 = NO WEATHER CLI02270
OAND ABSOLUTE HUMIDITY GE 10 GM/CU M./52H CLASS 17 = VIS LT 1 KM ANCLI02280
ED CEILING HEIGHT LT 300 M./53H CLASS 18 = VIS LT 3 KM AND CEILING CLI02290
FHEIGHT LT 1000 M./36H CLASS 19 = CEILING HEIGHT LT 300 M./37H CLASCLI02300
GS 20 = CEILING HEIGHT LT 1000 M./23H CLASS 21 = NO CEILING./36H CCLI02310
HASS 22 = ALL CONDITIONS COMBINED.) CLI02320
17 FORMAT(15,F9.1,13F7.1) CLI02330
CLI02340
CLI02350

END

PROGRAM AGAUS

PROGRAM AGAUS

REVISION DATE 22 JANUARY, 1982

PURPOSE:

TO CALCULATE EXTINCTION COEFFICIENTS, ETC., AND PRODUCE LEGENDRE EXPANSION COEFFICIENTS, PHASE FUNCTIONS AND (OPTIONALLY) SCATTERING FRACTIONS UNDER A VARIETY OF CONDITIONS AND AEROSOL DISTRIBUTIONS AT ONE OR MORE WAVELENGTHS. THE PHASE FUNCTION IS NORMALIZED TO 4 PI OMEGA ZERO AND MAY BE RENORMALIZED BY DIVISION BY THE APPROPRIATE CONSTANT(S).

***** INPUT *****

CARD 1 - IDENTIFIER - 80 ALPHA CHARACTERS

CARD 2 - INTEGER CONTROL PARAMETERS: NWAVE, NINDX, IW, IDSTP,
NRADI, IT, M RTE, IANG, IEO, NEOU
FORMAT (10I5)

NWAVE: IS THE NO. OF WAVELENGTHS, OR REL. HUMIDITY VALUES TO BE TREATED IN THIS RUN. SEE COMMENTS CIRCA READ OF WAVE, DWAVE, ETC.
N.B. NWAVE MUST BE .LE. 10 - TO CHANGE THE NUMBER OF WAVELENGTHS
CHANGE THE FIRST INDEX OF ARRAY OUT(I,J) TO AGREE WITH NWAVE.

NINDX: IS THE NBR OF AEROSOL COMPONENTS WHICH WILL HAVE DIFFERENT OPTICAL CONSTANTS, MASS DENSITIES OR MASS CONCENTRATIONS.

IW: =0 WILL SET THE REFRACTIVE INDEX OF THE AEROSOL EQUAL TO THAT OF WATER AT THE INPUT WAVELENGTH AND TEMP. IF IW .NE. 0 AND HANDEL'S GROWTH FACTOR IS ZERO (EMUA=0 - CARD 5), THEN THE INPUT REFRACTIVE INDEX (EMA,CAYA) WILL BE USED FOR THE AEROSOL. OTHERWISE, THE REFRACTIVE INDEX IS ADJUSTED PER HANDEL (SEE BELOW).

IDSTP: IDENTIFIES TYPE OF AEROSOL SIZE DISTRIBUTION TO BE USED.
NRADI: NO. OF PARTICLE RADII TO BE EXPECTED FOR IDSTP=0 OR 3.

THE INPUT VALUE OF NRADI IS IGNORED FOR IDSTP NOT ZERO OR 3.
NRADI MUST BE .LE. 1+2**J0IMCK(2) - C.F. BLOCK DATA

IT: IS THE NUMBER OF GAUSS-LEGENDRE ANGLES (ORDER OF EXPANSION)
IF ONLY EXTINCTION COEFFICIENTS, ETC. ARE DESIRED, I.E. NOT PHASE FUNCTIONS, THEN SET -IT- EQUAL TO ONE.

M RTE:=12345 WILL CAUSE PRINTS OF MIE EFFICIENCY FACTORS AT
EVERY VALUE OF PARTICLE RADIUS USED IN THE MIE CALCULATIONS;

SET M RTE = 0 IF SUCH PRINTS ARE NOT DESIRED.

IANG:=0 FOR COMPUTATIONS OF PHASE FN. AT -IT- GAUSS LEGENDRE QUADRATURE ANGLES; IANG=1 FOR COMPUTATIONS OF PHASE FN AT -IT- EQUALLY SPACED ANGLES BETWEEN 0 AND 180 DEGREES.

IANG=2 WILL ALLOW -IT- USER SUPPLIED ANGLES TO BE READ -
FORMAT (16F5.1). THIS REQUIRES AT LEAST ONE CARD OF TYPE 2A.

IF IANG.GT.0 NO LEGENDRE COEFFICIENTS WILL BE GENERATED.

IEO=1,2,3,4 WILL CONSTRUCT A PHASE FUNCTION FILE (ON NEOU).

IEO=1 65 PREDETERMINED ANGLES INDIVIDUAL WAVELENGTHS ONLY

IEO=2 65 PREDETERMINED ANGLES COMPOSITE WAVELENGTH ONLY

IEO=3 USER INPUT ANGLES INDIVIDUAL WAVELENGTHS ONLY

IEO=4 USER INPUT ANGLES COMPOSITE WAVELENGTH ONLY

IEO=5 65 PREDETERMINED ANGLES INDIVIDUAL & COMPOSITE WAVELENGTHS

THE COMPOSITE WILL BE THE LAST DATA

SET WRITTEN ON UNIT -NEOU-

FOR USER INPUT ANGLES SEE -IT- AND IANG ABOVE. THE COMPOSITE

VALUES ARE SIMPLE AVERAGES OVER THE NUMBER OF WAVELENGTHS.

THIS FILE WILL CONTAIN THE FOLLOWING INFORMATION:

1) ANGLES (65 MAX) - FORMAT(11(F6.2,1X))

2) NBR OF ANGLES, PHASE FUNCTION IDENTIFIER (=0= IMPLIED USER INPUT

IN EOSAEL), WAVELENGTH (KM), SINGLE SCATTERING ALBEDO, EXTINCTION

AND SCATTERING COEFFICIENTS IN INVERSE KM⁻¹

FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))

3) PHASE FUNCTION AT ANGLES SPECIFIED ABOVE. N.B. THE PHASE FUNCT

AS WRITTEN OUT HERE IS NORMALIZED TO 4 PI OMEGA ZERO; THE ROUTINE

IN EOSAEL WILL RENORMALIZE THE PHASE FUNCTION TO ONE.

FORMAT (6(E12.6,1X))

NEOU= UNIT NUMBER UPON WHICH EOSAEL PHASE FUNCTION IS TO BE STORED.

AGX00020
AGX00030
AGX00040
AGX00050
AGX00060
AGX00070
AGX00080
AGX00090
AGX00100
AGX00110
AGX00120
AGX00130
AGX00140
AGX00150
AGX00160
AGX00170
AGX00180
AGX00190
AGX00200
AGX00210
AGX00220
AGX00230
AGX00240
AGX00250
AGX00260
AGX00270
AGX00280
AGX00290
AGX00300
AGX00310
AGX00320
AGX00330
AGX00340
AGX00350
AGX00360
AGX00370
AGX00380
AGX00390
AGX00400
AGX00410
AGX00420
AGX00430
AGX00440
AGX00450
AGX00460
AGX00470
AGX00480
AGX00490
AGX00500
AGX00510
AGX00520
AGX00530
AGX00540
AGX00550
AGX00560
AGX00570
AGX00580
AGX00590
AGX00600
AGX00610
AGX00620
AGX00630
AGX00640
AGX00650
AGX00660
AGX00670
AGX00680
AGX00690
AGX00700

CARD 2A - USER SUPPLIED SET OF -IT- ANGLES FORMAT (16F5.1)
 16 VALUES PER CARD, MORE THAN 1 CARD MAY BE NEEDED.
 THIS CARD IS ONLY NEEDED WHEN IANG=2. AGX00710
 AGX00720

CARD 3 - DISTRIBUTION PARAMETERS; ONLY ONE TYPE PER RUN.
 FORMAT (6E12.6) - READ IN AGXP1 AGX00730
 AGX00740
 ***** ALL DIMENSIONS ARE IN MICRONS ***** AGX00750
 AGX00760
 AGX00770

TYPE 0. USER SUPPLIED - NRADI CARDS, ONE VALUE OF RADIUS AND
 NUMBER DENSITY PER CARD. IT IS SUGGESTED THAT DELTA BE
 INPUT NO GREATER THAN .001 IN ORDER TO FORCE THE MAXIMUM
 NUMBER OF RADII TO BE USED DUE TO THE POSSIBLE IRREGULAR
 NATURE OF THIS DISTRIBUTION.
 $R(I)$, $FF(I)$, $I = 1$, NRADI AGX00810

TYPE 1. LOG-NORMAL
 \bar{r}_{bar} , SIGMA, RLD, RHI AGX00820

TYPE 2. DOUBLE EXPONENTIAL
 RLD, RHI, CUE, A, B AGX00830

TYPE 3. DEIRMENDJIAN MODEL C
 - NO INPUT - AGX00840

TYPE 4. POWER LAW (JUNGE)
 RLD, RHI, CUE, A AGX00850

TYPE 5. MODIFIED GAMMA
 RLD, RHI, RC, ALF, GAM AGX00860

TYPE 6. MODIFIED GAMMA FOG MODEL
 RLD, RHI, RC, ALF, GAM, ELWC AGX00870

TYPE 7. POWER LAW
 VIS AGX00880

TYPE 8. CONTINENTAL BIMODAL
 - NO INPUT - AGX00890

TYPE 9. MARITIME BIMODAL
 - NO INPUT - AGX00900

TYPE 10. URBAN BIMODAL
 - NO INPUT - AGX00910

TYPE 11. USER SUPPLIED BIMODAL
 F_{DA} , R_{BAA} , SG_A , F_{DC} , R_{BARC} , SG_C AGX00920

TYPE 12. MARSHALL-PALMER RAIN MODEL
 RAIN AGX00930

CARD 4 - CONTROL PARAMETERS: FORMAT (6E12.6)
 WAVE, DWAVE, RELHUM, DENSH, TEMP, DELTA
 FOR LOOPING OVER RELATIVE HUMIDITY ADD
 NWAVE-1 CARDS CONTAINING RELHUM, TEMP - FORMAT(2E12.6)
 SEE DWAVE BELOW. AGX00940

WAVE: IS WAVELENGTH IN MICROMETERS. AGX00950

DWAVE: IS THE WAVELENGTH INCREMENT IN MICROMETERS. AGX00960

IF DWAVE IS LESS THAN 1.E-4, A SPECIAL CASE APPLIES USED FOR LOOPING OVER AGX00970

NWAVE VALUES OF RELHUM: THE FIRST TIME THIS CARD IS READ IT AGX00980

MUST CONTAIN WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA; THE SECOND AND AGX00990

SUBSEQUENT TIMES IT MUST ONLY HAVE RELHUM,TEMP ON IT. THIS AGX01000

ALSO REQUIRES REPETITION OF CARD 5. AGX01010

RELHUM: IS RELATIVE HUMIDITY IN PERCENT. AGX01020

DENSH: IS PARTICLE NUMBER PER CUBIC CENTIMETER. AGX01030

USER-SUPPLIED VALUE OF DENSH WILL BE IGNORED FOR IDSTP=3 OR GT 6 AGX01040

BECUSE THOSE DISTRIBUTIONS CARRY PRE-DETERMINED DENSITY VALUES. AGX01050

ALSO, IF DENSH IS LESS THAN 1E-4, THE PARTICLE NUMBER DENSITY AGX01060

WILL BE CALCULATED FROM MASS DENSITY AND MASS CONCENTRATION. AGX01070

TEMP: IS THE TEMPERATURE OF THE ATMOSPHERE IN DEGREES C. AGX01080

DELTA: IS THE CONVERGENCE CRITERION WITHIN A PARTICULAR SIZE AGX01090

RANGE INTERVAL: HALVING IS TERMINATED WHEN THE QUANTITY DEL AGX01100

IS LESS THAN DELTA. **N.B.** THE AMOUNT OF CPU TIME USED BY AGX01110

THIS PROGRAM IS CLOSELY CONNECTED WITH DELTA. THE SMALLER DELTA AGX01120

IS THE LARGER YOUR RUN TIME WILL BE. IT IS SUGGESTED THAT AGX01130

DELTA BE SET EQUAL TO .001 FOR MOST RUNS. AGX01140

CARD 5 - OPTICAL AND PHYSICAL DATA: FORMAT (4F10.5,E15.7)
 E_{MA} , C_{AYA} , E_{MUA} , ρ_{HOA} , CONC AGX01150

REPEAT NWAVE*NINDEX TIMES: IF IDSTP=6 THIS CARD IS NOT NEEDED. AGX01160

E_{MA} : IS THE REAL PART OF THE INDEX OF REFRACTION OF DRY AEROSOL. AGX01170

C_{AYA} : IS THE IMAGINARY PART OF REFRACTIVE INDEX FOR DRY AEROSOL. AGX01180

***** CAYA IS ASSUMED TO BE NEGATIVE **** AGX01190

***** DO NOT ENTER CAYA WITH A NEGATIVE SIGN !!!!! *****
 EMUA: IS HANEL'S GROWTH FACTOR (MU-BAR)/ACCRETION COEF. AGX01380
 RHUA: IS THE MASS DENSITY(SF. GRAV) OF DRY AEROSOL. AGY01400
 CONC: IS THE MASS CONCENTRATION(GM/CC) OF DRY AEROSOL. AGX01410
 AGX01420
 ***** END INPUT ***** AGX01430
 ***** MISCELLANEOUS INFO ***** AGX01450
 THE INPUT AND OUTPUT UNITS, ALONG WITH A EXTRA, CURRENTLY UNUSED, AGX01460
 UNIT (NUNIT) ARE ASSIGNED VALUES IN THE BLOCK DATA SUBROUTINE. AGX01470
 REL. HUMIDITY TREATMENT PER G. HANEL/1976 ADV. IN GEOPHYS. AGX01480
 AGX01490
 FOR DIMENSION SIZES REFER TO THE BLOCK DATA SUBROUTINE. THERE AGX01500
 ALSO IS A ERROR ROUTINE (DIMER) THAT CKS ON YOUR DIMENSIONS. AGX01520
 AGX01530
 SCATTERING FRACTIONS REQUIRE THAT NUNIT BE ASSIGNED AND A AGX01540
 SIMPLE CHANGE BE MADE IN SUBROUTINE AGXP3: FOR FURTHER INFO AGX01550
 REFER TO THAT SUBROUTINE. AGX01560
 AGX01570
 THE FUNCTION ATAN2(SQRT(1.-C(I)**2),C(I)) IS EQUIVALENT AGX01580
 TO ARCCOS(C(I)). AGX01590
 AGX01600
 ***** ACKNOWLEDGEMENTS ***** AGX01610
 THIS PROGRAM HAS BEEN CONSTRUCTED BY THE ATMOSPHERIC SCIENCES AGX01620
 LABORATORY AND NEW MEXICO STATE UNIVERSITY, DEPT OF PHYSICS. THE AGX01630
 FOLLOWING PEOPLE HAVE PARTICIPATED IN THIS UNDERTAKING: AGX01640
 AGX01650
 AGX01660
 DR. A.U. MILLER NMSU AGX01680
 DR. R.C. SHIRKEY ASL AGX01690
 DR. G.H. GOEDECKE NMSU AGX01700
 MR. E.J. BURLBAW NMSU AGX01710
 AGX01720
 THE POINT OF CONTACT IS R.C. SHIRKEY, ASL. PHONE (505) 678-5470 AGX01730
 OR AY 258-5470. AGX01740
 AGX01750

 REAL KEXTT,KSCAT,KBAKT AGX01760
 COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3) AGX01770
 COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514) AGX01780
 +,NRADI,PI, IDSTP,NKG,NHALV,NI AGX01790
 COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65) AGX01800
 COMMON /IO/ I0IN,I0OUT,NUNIT,IEO,NEOU AGX01820
 DIMENSION OUT(10,4),NTITLE(40) AGX01830
 ***** READ AND WRITE IDENTIFIER ***** AGX01840
 READ (I0IN,88) (NTITLE(I),I=1,40) AGX01850
 88 FORMAT (40A2) AGX01860
 WRITE (I0OUT,89) (NTITLE(I),I=1,40) AGX01870
 89 FORMAT(1H1,40A2//) AGX01880
 ***** READ INTEGER CONTROL PARAMETERS FOR THIS RUN ***** AGX01890
 READ (I0IN,103) NWAVE,NINDX,IW, IDSTP,NRADI,IT,MQRTE,IANG,IEO,NEOU AGX01900
 C ERROR CHECKS AGX01910
 IF (IT.LE.0) IT=1 AGX01920
 IF (IT.GT.JDIMCK(1)) CALL DIMER(1) AGX01930
 C IF (JDIMCK(1).LT.65) WRITE(I0OUT,129) AGX01940
 C CHECK FOR CONFLICTING EOSAEL OPTIONS AGX01950
 IF ((IEO.EQ.2.OR.IEO.EQ.5).AND.NWAVE.EQ.1) IEO=1 AGX01960
 IF ((IEO.EQ.4.OR.IEO.EQ.5).AND.NWAVE.EQ.1) IEO=3 AGX01970
 IF (IEO.GT.1.AND.IT.GT.65) GO TO 20 AGX01980
 C EOSAEL OPTION AGX01990
 IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) IT=65 AGX02000
 JDIMCK(3)=1+2**JDIMCK(2) AGX02010
 IF (IDSTP.GT.12) GO TO 1 AGX02020
 IF (NWAVE.EQ.0) NWAVE=1 AGX02030
 IF (IDSTP.EQ.12) IW=1 AGX02040
 IF (NINDX.LT.1.OR.IDSTP.EQ.6.OR.IDSTP.EQ.12) NINDX=1 AGX02050
 WRITE (I0OUT,104) NWAVE,NINDX,IW, IDSTP,NRADI,IT,MQRTE,IANG,IEO,NEOU AGX02060
 IF (IW.EQ.0) WRITE (I0OUT,122) AGX02070

```

C   INITIALIZE QUANTITIES USED IN SUMMATIONS          AGX02080
DO 2 I=1,IT                                         AGX02090
OLT(I)=0.                                            AGX02100
2 PSUMT(I)=0.E0                                      AGX02110
WAVAVG=0.                                           AGX02120
ALBDOT=0.                                           AGX02130
KEXTT=0.E0                                           AGX02140
KSCAT=0.E0                                           AGX02150
KBAKT=0.E0                                           AGX02160
CATTN=0.E0                                           AGX02170
ITT=IT-1                                           AGX02180
PI=3.1415926535898E+00                            AGX02190
IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) GO TO 3    AGX02200
IF ((IANG.EQ.1).OR.(IANG.EQ.2)) GO TO 3
WHEN IANG=0 ROUTINE GUSET IS CALLED TO SET-UP THE ABSCISSAE AND
WEIGHTS USED FOR CALCULATING THE PHASE-FUNCTION AT -IT- POINTS
USED FOR NUMERICAL INTEGRATION VIA GAUSS-LEGENDRE QUADRATURE AND
THE PHASE FUNCTION EXPANSION COEFS, OLK(), AGX02210
THE WEIGHTS ARE PLACED IN THE ARRAY W(), AND THE COSINES OF THE
ANGLES ARE PLACED IN THE ARRAY C().                AGX02220
CALL GUSET(IT)                                     AGX02230
IF (ITT.LT.3) ITT=3                                AGX02240
GO TO 7
3 CALL ANGLE (PI,IANG,IT)                           AGX02250
SUBROUTINE ANGLE IS CALLED WHEN IANG=1 OR 2 TO SET UP THE
ANGLES AT WHICH PHASE FUNCTIONS WILL BE CALCULATED. ANGLES
GO INTO ARRAY W() AND COSINES IN C().               AGX02260
CONTINUE
4 WRITE ANGLES FOR EOSAEL DATA FILE                 AGX02270
IF (IEO.LE.0) GO TO 21                             AGX02280
5 DO 22 I=1,IT                                     AGX02290
C(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI        AGX02300
ITP1=IT+1                                         AGX02310
IF (ITP1.GT.JDIMCK(1)) ITP1=IT                   AGX02320
IF (IT.LT.65.AND.JDIMCK(1).GT.65) C(ITP1)=999.99  AGX02330
WRITE (NEOU,125) (C(I),I=1,ITP1)                  AGX02340
DO 23 I=1,IT                                     AGX02350
C(I)=COS(C(I)*PI/180.)                           AGX02360
CONTINUE
6 DETERMINE DETAILS OF AEROSOL SIZE-DISTRIBUTION VIA AGXP1
CALL AGXP1(DENS,FSUM,VOL,JDIMCK)                  AGX02370
IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS         AGX02380
DRYVOL IS THE AVERAGE VOLUME OF THE DRY AEROSOL PARTICLES IN
CUBIC MICROMETERS.                                AGX02390
DRYVOL=VOL                                         AGX02400
C *** READ INPUT PARAMETERS ***
READ (IOIN,105) WAVE,DWAVE,RELHUM,DENS,TEMP,DELTA
IF (NWAVE.EQ.1) DWAVE=0.E0                          AGX02410
WRITE (IOUT,106) WAVE,DWAVE,RELHUM,DENS,TEMP,DELTA
IF (NINDX.GT.1) WRITE (IOUT,107) NINDEX           AGX02420
IF ((DWAVE.LT.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,108) NWAVE
IF ((DWAVE.GE.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,109) NWAVE
IF (DENS.LT.1E-04) WRITE (IOUT,110)
ENWAV=FLOAT(NWAVE)
IF (DWAVE.LT.1.E-4) GO TO 8
WAVE=WAVE-DWAVE                                     AGX02430
8 DO 9 NWV=1,NWAVE                                    AGX02440
IF (DWAVE.GT.1.E-4) GO TO 10
IF (NWV.EQ.1) GO TO 11
READ (IOIN,105) RELHUM,TEMP                         AGX02450
GO TO 11
9 WAVE=WAVE+DWAVE                                     AGX02460
10 VOL=DRYVOL                                         AGX02470
11 DETERMINE WHETHER THE USER SUPPLIED PARTICLE NUMBER DENSITY DENS
SHOULD BE OVERIDDEN BECAUSE THE CHOSEN IDSTP CASE HAS FIXED
PARAMETERS, AND/OR IF NUMBER DENSITIES ARE TO BE CALCULATED LATER
FROM THE AVG PARTICLE VOLUME, MASS DENSITY, AND MASS CONCENTRATION
LLLL=0                                              AGX02480
IF (IDSTP.EQ.6) GO TO 12
IF (IDSTP.EQ.3.OR.IDSTP.GE.7) LLLL=1              AGX02490

```

```

IF (LLLL.EQ.1) GO TO 12          AGX027800
IF (DENS.H.E.1.E-4) GO TO 12      AGX027900
LLLL=1                           AGX028000
DENS=DENSH                         AGX028100
C 12 CONTINUE                      AGX028200
RESTRICT RELATIVE HUMIDITY TO MAX OF 99 PERCENT.    AGX028300
IF (RELHUM.GE.99.E+00) RELHUM=99.0E+00      AGX028400
WRITE (IOUT,111) RELHUM,WAVE           AGX028500
IF (DENS.EQ.0.0) DENS=1.E+00            AGX028600
GNU=1.0E+04/WAVE                     AGX028700
IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) DENS=ELWC      AGX028800
AGX028900
C DENS IS USED AS AN ALIAS TO PASS ELWC TO ROUTINE AGXP2.
CALL AGXP2(RELHUM,CTSUM,CSSUM,CRSUM,VOL,TMASS,DENS,WATTN,TEMP,
1 LSTAR,NINDEX,IW,OM2,LLLL,IT,WAVE,EM,CAY,EMM,MQRTE,PFNZRO)  AGX029000
1 LMAX=3*IFIX(2.E+0*PI*EMM*R(NRADI)/WAVE)        AGX029100
IF (LMAX.GT.IT) WRITE (IOUT,112) LMAX,IT       AGX029200
13 CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDEX,WAVE,EM,CAY,EMM,IT,0,
+IANG)                                     AGX029300
C SUM QUANTITIES OVER INDEX NWV.             AGX029400
DO 14 IK=1,IT                         AGX029500
OLT(IK)=OLT(IK)+OL(IK)                AGX029600
14 PSUMT(IK)=PSUMT(IK)+PSUM(IK)        AGX029700
AGX029800
C ALBDOT BECOMES THE TOTAL SINGLE SCATTERING ALBEDO      AGX030000
C KEXTT BECOMES THE TOTAL EXTINCTION COEF. (PER KILOMETER)  AGX030100
C KSCAT BECOMES THE TOTAL SCATTERING COEF. (PER KM)        AGX030200
C KBAKT BECOMES THE TOTAL BACK-SCATTERING(RADAR) COEF (PER KM)  AGX030300
C ARRAY OUTC() HOLDS SOME QUANTITIES FOR LATER PRINTOUTS   AGX030400
ALBDO=CSSUM/CTSUM                     AGX030500
ALBDOT=ALBDOT+ALBDO                  AGX030600
KEXTT=KEXTT+CTSUM                    AGX030700
KSCAT=KSCAT+CSSUM                  AGX030800
KBAKT=KBAKT+CRSUM                  AGX030900
CATTN=CATTN+WATTN                 AGX031000
WAVAVG=WAVAVG+WAVE                 AGX031100
OUT(NWV,1)=WAVE                     AGX031200
OUT(NWV,2)=RELHUM                  AGX031300
OUT(NWV,3)=TMASS*1.E5               AGX031400
OUT(NWV,4)=CTSUM                   AGX031500
IF ((NWAVE.GT.1).AND.(DWAVE.GE.1.E-04)) WRITE (IOUT,113) NWV  AGX031600
IF ((NWAVE.GT.1).AND.(DWAVE.LT.1.E-04)) WRITE (IOUT,114) NWV  AGX031700
C EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH, SINGLE SCATTERING  AGX031800
C ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR      AGX031900
C INDIVIDUAL WAVELENGTHS.                                         AGX032000
C IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,127)        AGX032100
C + IT,WAVE,ALBDOT,CTSUM,CSSUM                                AGX032200
C EOSAEL OPTION: WRITE PHASE FUNCTION FOR INDIVIDUAL WAVELENGTHS.  AGX032300
C IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,128)        AGX032400
C + (PSUM(I),I=1,IT)                                           AGX032500
9  CONTINUE                      AGX032600
C END OF NWAVE LOOP          AGX032700
IF (NWAVE.LE.1) GO TO 19          AGX032800
C DIVIDE BY NBR OF VALUES OF NWV TO GET AVERAGED RESULTS  AGX032900
DO 16 I=1,IT                     AGX033000
OL(I)=OLT(I)/ENWAV                AGX033100
PSUM(I)=PSUMT(I)/ENWAV           AGX033200
16 CONTINUE                      AGX033300
ALBDOT=ALBDOT/ENWAV              AGX033400
KEXTT=KEXTT/ENWAV                AGX033500
KSCAT=KSCAT/ENWAV                AGX033600
KBAKT=KBAKT/ENWAV                AGX033700
CATTN=CATTN/ENWAV                AGX033800
WAVAVG=WAVAVG/ENWAV              AGX033900
WRITE (IOUT,117) NWAVE            AGX034000
WRITE (IOUT,118)                  AGX034100
DO 18 J=1,NWAVE                  AGX034200
18 WRITE (IOUT,119) (OUT(J,JJ),JJ=1,4)      AGX034300
WRITE (IOUT,123) KEXTT,KSCAT,KBAKT,CATTN,ALBDOT      AGX034400
CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDEX,WAVE,EM,CAY,EMM,IT,1,
+IANG)                                     AGX034500
C EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH ,SINGLE SCATTERING  AGX034600

```

```

C ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR          AGX03480
C COMPOSITE VALUES.                                                 AGX03490
C IF (IE0.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,127)           AGX03500
C + IT,WAVAVG,ALBDDOT,KEXTT,KSCAT                                     AGX03510
C EOSAEL OPTION; WRITE COMPOSITE PHASE FUNCTION.                   AGX03520
C IF (IE0.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,128)           AGX03530
C + (PSUM(I),I=1,IT)                                                 AGX03540
C GO TO 19                                                       AGX03550
1 1 WRITE (IOUT,120) IDSTP                                         AGX03560
19 WRITE (IOUT,126) STOP                                           AGX03570
20 WRITE (IOUT,124) STOP                                           AGX03580
20 FORMAT (10I5)                                                 AGX03590
104 FORMAT(1H,78HINTEGER CONTROL PARAMETERS: NWAVE NINDEX IW IDSTP NRAAGX03620
+DI IT MORTÉ IANG IEO NEOU/,1X,29X,2(I2,4X),I1,3X,I2,          AGX03630
+3X,I3,1X,I3,1X,15,1X,15,2(2X,I2))                                AGX03640
105 FORMAT (6E12.6)                                              AGX03650
106 FORMAT(1H,/,17H INPUT PARAMETERS/,1X,6X,9HWAVE = ,E12.6,8H MICRAGX03660
+ONS/,1X,6X,9HDWAVE = ,E12.6,8H MICRONS/,1X,6X,9HRELHUM = ,E12.6, AGX03670
+8H PERCENT/,1X,6X,9HDENSH = ,E12.6,13H PARTICLES/CC/,1X,6X,      AGX03680
+9HTEMP = ,E12.6,6H DEG C/,1X,6X,19HDELTA (CONVERGENCE ,        AGX03690
+13HCITERION) = ,E12.6)                                         AGX03700
107 FORMAT (/,1H,29HLOOPING OPTION IN EFFECT FOR ,I2,             AGX03710
+19H AEROSOL COMPONENTS)                                         AGX03720
108 FORMAT (/,1H,39HRELATIVE HUMIDITY OPTION IN EFFECT FOR ,I2,       AGX03730
+7H VALUES)                                                 AGX03740
109 FORMAT (/,1H,40HWAVELENGTH LOOPING OPTION IN EFFECT FOR ,I2,       AGX03750
+12H WAVELENGTHS)                                              AGX03760
110 FORMAT(1H,52H*** PARTICLE NUMBER DENSITY WILL BE CALCULATED FROMAGX03770
+41H MASS DENSITY AND MASS CONCENTRATION ***)                  AGX03780
111 FORMAT (1H1,/,1X,33HRELATIVE HUMIDITY FOR THIS RUN = ,F6.2,       AGX03790
+25H PERCENT. WAVELENGTH = ,F10.3,8H MICRONS/)                 AGX03800
112 FORMAT (/,49H *** WARNING *** OPTIMAL PF EXPANSION ORDER OF ,I3,   AGX03810
+ 22H EXCEEDS INPUT IT = ,I3,24H. PF VALUES SHOULD BE , AGX03820
+ 15HUSED CAUTIOUSLY/)                                         AGX03830
113 FORMAT (1H //,1X,40(1H*),3X,31HEND OF WAVELENGTH CYCLE NUMBER ,I3,AGX03840
+3X,40(1H*))                                                 AGX03850
114 FORMAT (1H //,1X,40(1H*),3X,38HEND OF RELATIVE HUMIDITY CYCLE NUMBAAGX03860
+ER ,I3,3X,40(1H*))                                         AGX03870
117 FORMAT(1H1,/,47H SUMMARY OF RESULTS FOR THIS RUN AVERAGED OVER , AGX03880
+I2,30H WAVELENGTHS) ARE AS FOLLOWS://)                         AGX03890
118 FORMAT(1H ,4X,48HWAVELENGTH REL.HUMIDITY AEROSOL MASS KC,AGX03900
+ 11HEXTINCTION),/1X,29H (MICROMETERS) (PERCENT)               AGX03910
+ .6H (GM,25H/(SQ.CM-KM)) (PER KM)//)                          AGX03920
119 FORMAT (2F15.6,1P2E16.5)                                       AGX03930
120 FORMAT (//13H *** IDSTP = ,I5,35H IS ILLEGAL. EXECUTION TERMINATEDAGX03940
+ 2H* //)                                                 AGX03950
122 FORMAT (/,1X,23H*** WATER ONLY CASE ***)                      AGX03960
123 FORMAT (/,20H EXTINCTION COEF. = ,5X,1PE13.7,9H (PER KM),/,     AGX03970
120H SCATTERING COEF. = ,8X,1PE13.7,9H (PER KM),/,              AGX03980
225H BACK-SCATTERING COEF. = ,3X,1PE13.7,9H (PER KM),/,          AGX03990
321H ATTENUATION COEF. = ,7X,1PE13.7,13H SQ-METERS/MG,/,          AGX04000
428H SINGLE SCATTERING ALBEDÓ = ,1PE13.7,/)                     AGX04010
124 FORMAT(1H ,58H*** MORE THAN 65 ANGLES FOR EOSAEL OPTION - PGM TERMAGX04020
+INATED)                                                 AGX04030
125 FORMAT (11(F6.2,1X))                                         AGX04040
126 FORMAT(1H1)                                                 AGX04050
127 FORMAT(12,1X,2H00,1X,F5.2,1X,F8.6,1X,2(E12.6,1X))          AGX04060
128 FORMAT(6(E12.6,1X))                                         AGX04070
129 FORMAT(1H ,23H**** AGAUS WARNING ****,/,1X,
+ 37HTHE ARRAY W IS ASSIGNED 65 VALUES IN ,/,1X,
+ 43HBLOCK DATA WHICH IS LARGER THAN ARRAY SIZE ,/,1X,
+ 47HYOU MAY BE ClobberING INSTRUCTIONS AND/OR DATA ,/)          AGX04080
END                                                       AGX04090
                                                AGX04100
                                                AGX04110
                                                AGX04120

```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND W5--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-8 AND ANCILLARY CODES AGAUS AND --ETC(1)
FEB 82 R G STEINHOFF

UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU

NL

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

5 - 6

```

SUBROUTINE AGXP1 (DENS,FSUM,VOL,JDIMCK)
COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)          AGA00010
+ NRADI,PI, IDSTP,NKG,NHALV,NI                         AGA00020
COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                   AGA00030
DIMENSION JDIMCK(3)                                     AGA00040
EXTERNAL GAMMA                                         AGA00050
WRITE (IOUT,2)                                         AGA00060
C      CHOOSE AND SET UP PARTICLE SIZE DISTRIBUTION      AGA00070
IF (IDSTP.NE.0) GO TO (12,15,16,17,19,17,20,21,22,24,26),IDSTP AGA00080
C**   TYPE 0: ARBITRARY USER-SUPPLIED DISTRIBUTION. NRADI VALUES OF
C      R(J) AND FF(J) MUST BE GIVEN, ONE PER CARD, AND READ IN ORDER
C      FROM SMALLEST RADIUS, RLO TO THE LARGEST.
C      NRADI MUST BE LESS THAN OR EQUAL TO 1+2**JDIMCK(2)    AGA00090
C      WRITE (IOUT,?)                                     AGA00140
C      IF (NRADI.GT.JDIMCK(3)) CALL DIMER(2)             AGA00160
DO 9 J=1,NRADI                                         AGA00170
 9 READ (IOIN,1) R(J),FF(J)                           AGA00190
  RLO=R(1)
  DO 10 J=1,NRADI,5
  JK=J+4
  IF(JK.GT.NRADI)JK=NRADI
 10 WRITE (IOUT,11)(R(K),FF(K),K=J,JK)               AGA00200
  WRITE (IOUT,11)                                     AGA00210
  FF(NRADI+1)=FF(NRADI)                            AGA00220
  RR(1)=RLO                                         AGA00230
  RR(2)=R(NRADI)                                    AGA00240
  MIN=0                                              AGA00250
  GO TO 28                                         AGA00260
C**   TYPE 1: LOG-NORMAL DISTRIBUTION                  AGA00270
C      12 READ (IOIN,1) RBAR,SIGMA,RLO,RHI              AGA00280
C      SIGMA IS STANDARD DEVIATION, NOT LN(SIGMA)       AGA00290
  SIGIN=SIGMA                                         AGA00300
  SIGMA=ALOG(SIGMA)                                  AGA00310
  A=ABS(1.E0/(2.5066283E0*SIGMA))                 AGA00320
  IF ((RHI-RLO).LE.1.E-4) GO TO 13                  AGA00330
  RR(1)=RLO                                         AGA00340
  RR(3)=RHI                                         AGA00350
  GO TO 14                                         AGA00360
 13 RR(1)=RBAR*EXP(-4.E0*SIGMA)                    AGA00370
  RR(3)=RBAR*EXP(4.E0*SIGMA)                        AGA00380
 14 RR(2)=RBAR                                         AGA00390
  MIN=1                                              AGA00400
  WRITE (IOUT,3) RBAR,SIGIN,RLO,RHI                 AGA00410
  AVOL=4.18879E0*(RBAR**3.E0)*EXP(4.5E0*SIGMA*SIGMA) AGA00420
C      HERE AND ELSEWHERE, AVOL IS THE VOLUME OBTAINED VIA
C      ANALYTICAL INTEGRATION OVER THE LIMITS RLO =0 TO RHI =
C      INFINITY: THAT CAN ONLY BE DONE FOR A FEW IDSTP CASES. AGA00430
  GO TO 28                                         AGA00440
C**   TYPE 2: DOUBLE EXPONENTIAL F(R)=CUE*A*EXP(-A*R)+(1-CUE)*B*EXP(-B*R). AGA00450
C      RESTRICTIONS: RHI.GT.RLO, B.GT.A.GE.0, 0.LE.CUE.LE.1.0. AGA00460
 15 READ (IOIN,1) RLO,RHI,CUE,A,B                  AGA00470
  WRITE (IOUT,4) RLO,RHI,CUE,A,B                   AGA00480
  RR(1)=RLO                                         AGA00490
  RR(3)=RHI                                         AGA00500
  RR(2)=0.5E0*(RLO+RHI)                           AGA00510
  MIN=1                                              AGA00520
  GO TO 28                                         AGA00530
C**   TYPE 3: DEIRMENDJIAN MODEL C. F(R) = 1.0, RLO.LE.R.LE.4*DELRD, AGA00540
C      F(R)=(4*DELRD/R)**4, R.GE.(4*DELRD)            AGA00550
C      NRADI IS READ IN EARLIER IN THE MAIN PROGRAM. AGA00560
 16 DENS=1.378E+04                                 AGA00570
  DELRD=0.02E0                                       AGA00580
  RLO=0.02E0                                         AGA00590
  RHI=RLO+DELRD* FLOAT(NRADI-1)                     AGA00600
  RR(1)=RLO                                         AGA00610
  RR(3)=RHI                                         AGA00620
  MIN=1                                              AGA00630
  RR(2)=RLO+4.E0*DELRD                           AGA00640
  GO TO 28                                         AGA00650
C**   TYPE 4 AND TYPE 7: POWER LAW. F(R) = CUE*R**-A    AGA00660

```

```

C   17 RLO,LE,R,LF,RHI; VIS=VISIBILITY IN KILOMETERS.          AGA00720
C** 17 IF (IDSTP.EQ.4) GO TO 18                                AGA00730
C** TYPE 7 PRESCRIBED PARAMETER.                               AGA00740
C      READ (IOIN,1) VIS                                       AGA00750
C      RLO=0.1E0                                              AGA00760
C      RHI=15.E0                                             AGA00770
C      CUE=30.E0                                             AGA00780
C      R=4.E0                                                AGA00790
C      DENS=11.E0**((5.E0-ALOG10(VIS)))                      AGA00800
C** 18 TYPE 4 PRESCRIBED PARAMETERS;                         AGA00810
C      IF (IDSTP.EQ.4) READ (IOIN,1) RLO,RHI,CUE,A           AGA00820
C      WRITE (IOUT,5) RLO,RHI,CUE,A,VIS                      AGA00830
C      RRC(1)=RLO                                           AGA00840
C      RRC(3)=RHI                                           AGA00850
C      RRC(2)=<0.5E0*(RLO**(-A)+RHI**(-A))**(-1.E0/A)      AGA00860
C      MIN=1                                               AGA00870
C      GO TO 28                                            AGA00880
C** 19 TYPE 5: MODIFIED GAMMA/GENERALIZED KHIRGIAN-MAZIN    AGA00890
C      F(R) = (R**ALF)*EXP(-ALF*((R/RC)**GAM)/GAM)        AGA00900
C      RLO,LE,R,LE,RHI                                     AGA00910
C** 20 TYPE 6: SPECIAL CASE FOR WATER FOGS OR CLOUDS,       AGA00920
C      IN WHICH CASE ELWC IS LIQUID WATER CONTENT            AGA00930
C      IN GRAMS PER CUBIC CENTIMETER:                        AGA00940
C      ELWC IS IGNORED IF IDSTP = 5.                          AGA00950
C      READ (IOIN,1) RLO,RHI,RC,ALF,GAM,ELWC                AGA00960
C      IF (IDSTP.EQ.6) DENS=ELWC                           AGA00970
C      WRITE (IOUT,6) RLO,RHI,RC,ALF,GAM                   AGA00980
C      RRC(1)=RLO                                           AGA00990
C      RRC(2)=RC                                            AGA01000
C      RRC(3)=RHI                                           AGA01010
C      MIN=1                                               AGA01020
C      B=ALF/(GAM*RC**GAM)                                 AGA01030
C      AVOL=4.1888*B**(-3./GAM)*GAMMA((ALF+4.)/GAM)/GAMMA((ALF+1.)/GAM) AGA01040
C      GO TO 28                                            AGA01050
C** 21 TYPES 8,9,10: BIMODAL LOG-NORMAL DISTRIBUTIONS.     AGA01060
C      METHOD BELOW VALID FOR RBARC*EXP(-SGA),GT,RBARA*EXP(SGA) AGA01070
C** 22 TYPE 8: CONTINENTAL BIMODAL.                         AGA01080
C      FOA=4.E03                                           AGA01090
C      FOC=2.1E0                                           AGA01100
C      SGA=0.74E0                                           AGA01110
C      SGC=0.81E0                                           AGA01120
C      RBARA=0.03E0                                         AGA01130
C      RBARC=0.4E0                                           AGA01140
C      GO TO 23                                            AGA01150
C** 23 TYPE 9: MARITIME BIMODAL.                           AGA01160
C      FOA=4.E02                                           AGA01170
C      FOC=3.8E0                                           AGA01180
C      SGA=0.68E0                                           AGA01190
C      SGC=0.74E0                                           AGA01200
C      RBARA=0.05E0                                         AGA01210
C      RBARC=0.65E0                                         AGA01220
C      GO TO 23                                            AGA01230
C** 24 TYPE 10: URBAN BIMODAL.                            AGA01240
C      FOA=2.E04                                           AGA01250
C      FOC=0.6E0                                           AGA01260
C      SGA=0.63E0                                           AGA01270
C      SGC=.77E0                                           AGA01280
C      RBARA=0.04E0                                         AGA01290
C      RBARC=0.63E0                                         AGA01300
C      CALCULATE RADII FOR TYPES 8,9,10.                  AGA01310
C      RRC(1)=RBARA*EXP(-4.E0*ABS(SGA))                  AGA01320
C      RRC(2)=RBARA                                         AGA01330
C      RRC(3)=RBARA*EXP(4.E0*ABS(SGA))                  AGA01340
C      RRC(4)=RBARC*EXP(-4.E0*ABS(SGC))                  AGA01350
C      RRC(5)=RBARC*EXP(4.E0*ABS(SGC))                  AGA01360
C      MIN=2                                               AGA01370
C      DO 60 J=1,4                                         AGA01380
C      DO 60 I=1,4                                         AGA01390
C      IF (RRC(I+1).GT.RR(I)) GO TO 60                  AGA01400
C      HH=RR(I)                                           AGA01410

```

```

      RRC(1)=RRC(1+1)          AGA01420
      RRC(1+1)=HH              AGA01430
60    CONTINUE                AGA01440
      GO TO 28                AGA01450
C**  TYPE 11:                AGA01460
C     USER SUPPLIED BIMODAL CASE: FOA AND FOC ARE THE NUMBER DENSITIES
C     FOR THE ACCUMULATION (SMALLER RBAR) AND COARSE MODES,
C     RESPECTIVELY, IN PARTICLES PER CUBIC CENTIMETER;
C     SGA IS STD.DEVIATION FOR MODE A ** NOT LN(SIGMA) ***
C     SGC IS STD.DEVIATION FOR MODE C ** NOT LN(SIGMA) ***
C     *** NOTE, HOWEVER, THAT SGA AND SGC ARE THE LOGS OF THE
C     STANDARD DEVIATIONS IN THE PRE-CODED CASES TYPE 8-10.    AGA01470
      24 READ (I0IN,1) FOA,RBARA,SGA,FOC,RBARC,SGC          AGA01480
         WRITE (I0UT,25) FOA,RBARA,SGA,FOC,RBARC,SGC          AGA01490
         SGA=ABS ALOG(SGA))                                AGA01500
         SGC=ABS ALOG(SGC))                                AGA01510
         GO TO 23                                            AGA01520
C**  TYPE 12: MARSHALL-PALMER RAIN MODEL.          AGA01530
C     C.F. MASON, PHYSICS OF CLOUDS, CH. ON RADAR METEOROLOGY.
C     INPUT PARAMETER RAIN IS RAIN RATE IN MILLIMETERS/HOUR.
C     ** EMA, CAYA, AND RHQA ARE REQUIRED FOR THIS DISTRIBUTION.    AGA01540
      26 READ (I0IN,1) RAIN          AGA01550
         ENZERO=0.08E0          AGA01560
         CAPL=41.E0*RAIN**(-0.21E0)          AGA01570
         DENS=ENZERO/CAPL          AGA01580
         AVOL=PI*((CAPL**(-3.E0))*1.E12          AGA01590
         CONVERT UNITS FROM CM-4 TO (CM-3)*(MICROMETERS**(-1));
         THE FACTOR OF 2 CONVERTS THE M-P FORMULA FROM DIAMETER-DATA TO    AGA01600
         RADIUS BASED FORM.          AGA01610
         ENZERO=2.E-4*ENZERO          AGA01620
         CAPL=2.E-4*CAPL          AGA01630
         MIN=0          AGA01640
         RRC(1)=1.E-4          AGA01650
         RRC(2)=2500.E0          AGA01660
         WRITE (I0UT,27) RAIN,DENS          AGA01670
C     THE NEXT BLOCK IS COMMON TO ALL DISTRIBUTIONS.          AGA01680
C     IT SETS THE NMAX VALUES OF RADIUS, R(KK).          AGA01690
      28 MAX=JDIMCK(2)          AGA01700
         NHALV=MAX-MIN          AGA01710
         NMAX=1+2**MAX          AGA01720
         NI=2**MIN          AGA01730
         IF (NMAX.GT.JDIMCK(3).OR.NI.GT.JDIMCK(3)) CALL DIMER(3)          AGA01740
         NLAST=NI+1          AGA01750
         NKG=2**NHALV          AGA01760
         ENKG= FLOAT(NKG)
         IF (IDSTP.EQ.0) GO TO 30          AGA01770
DO 29 I=1,NI          AGA01780
         DR(I)=RRC(I+1)-RRC(I)          AGA01790
DO 29 K=1,NKG          AGA01800
         KK=(I-1)*NKG+K          AGA01810
         29 R(KK)=RRC(I)+FLOAT(K-1)*DR(I)/ENKG          AGA01820
         R(NMAX)=RRC(NLAST)          AGA01830
C     BRANCH AGAIN CALCULATE THE DIFFERENT F(R) ON THE NMAX POINTS R(K)          AGA01840
         GO TO (31,33,35,38,41,41,39,43,43,43,43,46),IDSTP          AGA01850
C**  TYPE 0; ARBITRARY          AGA01860
C     INTERPOLATE TO EQUAL INCREMENTS OVER RADII          AGA01870
30    DELR=(R(NRADI)-RL0)/ENKG          AGA01880
         F(1)=FF(1)
         NMAXM1=NMAX-1          AGA01890
         DO 64 KK=1,NMAXM1
         RADUS=RL0+DELR*FLOAT(KK)
         DO 62 J=1,NRADI
         K=J
         IF (R(J).GE.RADUS) GO TO 61
62    CONTINUE
61    CONTINUE
         F(KK+1)=(RADUS-R(K-1))*(FF(K)-FF(K-1))/          AGA01900
         (R(K)-R(K-1))+FF(K-1)
64    CONTINUE
         DO 65 I=1,NI          AGA01910

```

```

DR(I)=RR(I+1)-RR(I)
DO 65 K=1,NKG
  KK=(I-1)*NKG+K
  R(KK)=RR(I)+( FLOAT(K-1))*DR(I)/ENKG
  R(NMAX)=RR(NLAST)
  GO TO 48
C** TYPE 1: LOG NORMAL
  31 DEN=2,E0*SIGMA*SIGMA
  DO 32 KK=1,NMAX
    GNUM=ALOG(R(KK))/RBAR
  32 F(KK)=EXP(-GNUM*DEN)*A/R(KK)
  GO TO 48
C** TYPE 2: DOUBLE EXPONENTIAL
  33 DO 34 KK=1,NMAX
    FKK=(1,E0-CUE)*B*EXP(-B*R(KK))
  34 F(KK)=FKK+CUE*A*EXP(-A*R(KK))
  GO TO 48
C** TYPE 3: DEIRMENDJIAN MODEL C.
  35 DU 36 KK=1,NMAX
  36 F(KK)=1,E0
  NKG1=NKG+1
  DO 37 KK=NKG1,NMAX
  37 F(KK)=(RR(2)/R(KK))**4,E0
  GO TO 48
C** TYPES 4 AND 7: POWER LAW
  38 GO TO 39
  39 DU 40 KK=1,NMAX
  40 F(KK)=CUE*R(KK)**(-A)
  GO TO 48
C** TYPE 5 AND TYPE 6: MODIFIED GAMMA
  41 DO 42 KK=1,NMAX
  42 F(KK)=(EXP(-B*R(KK)**GAM)*R(KK)**ALF
  GO TO 48
C** TYPES 8, 9, 10, 11: BIMODAL LOG-NORMAL DISTRIBUTIONS
  43 DENA=2,E0*SGA*SGA
  DENC=2,E0*SGC*SGC
  FAA=F0A/SGA
  FCC=F0C/SGC
  DO 44 KK=1,NMAX
  GNUMA=ALOG(R(KK))/RBARA
  GNUMC=ALOG(R(KK))/RBARC
  FA=FAA*EXP(-GNUMA+GNUMA/DENA)
  FC=FCC*EXP(-GNUMC+GNUMC/DENC)
  44 F(KK)=(FA+FC)/R(KK)
  DENS=F0A+F0C
  WRITE (IOUT,45) DENS
  VOLA=4.18879E0*(RBARA**3,E0)*EXP(4.5E0*SGA*SGA)*F0A
  VOLC=4.18879E0*(RBARC**3,E0)*EXP(4.5E0*SGC*SGC)*F0C
  AVOL=(VOLA+VOLC)/DENS
  GO TO 48
C** TYPE 12: MARSHALL-PALMER RAIN MODEL
  46 DO 47 KK=1,NMAX
  47 F(KK)=ENZERO*EXP(-CAPL*R(KK))
C CALCULATE NORMALIZED F(KK) AND SOME DRY VOLUMES USING ALL NMAX
C VALUES OF RADII.
C VOL=AVERAGE PARTICLE VOLUME IN A DISTRIBUTION. THE
C NORMALIZATION AND FURTHER VOLUMES ARE RECALCULATED LATER
C BY THE HALVING INTEGRATION METHOD.
  48 FSUM=0,E0
  IF(F(1).LT.,0,E0)F(1)=0.0E0
  DO 49 J=2,NMAX
  IF(F(J).LT.,0,E0)F(J)=0,E0
  49 FSUM=FSUM+0.5E0*(F(J)+F(J-1))*(R(J)-R(J-1))
  DO 50 J=1,NMAX
  F(J)=F(J)/FSUM
  WRITE (IOUT,8) FSUM
  NRADI=NMAX
  IF (IDSTP.EQ.1,OR.IDSTP.EQ.5,OR.IDSTP.GE.8) WRITE (IOUT,51) AVOL
  VOL=0,E0
  DO 52 J=2,NMAX

```

```

52 VOL=VOL+2.0944E0*(F(J)*R(J)**3.E0+F(J-1)*R(J-1)**3.E0)*(R(J)-R(J-
1))
1 WRITE (IOUT,53) VOL
C THE VOLUME PER PARTICLE CALCULATED HERE IS OBTAINED USING
ALL AVAILABLE (NMAX VALUES) VALUES FOR THE PARTICLE RADII.
C WRITE (IOUT,54)
DO 56 INT=1,NI
INF=INT+1
56 WRITE (IOUT,55) INT,RR(INT),RR(INF)
1 FORMAT (6E12.6,I3)
2 FORMAT (1H, //24H AEROSOL PARAMETERS ARE )
3 FORMAT (1H,24X,6H RBAR=,E12.6,5X,7HSIGMA= ,E12.6,7H RLO = ,E12.6,AGA02660
+ ,E12.6/) AGA02670
4 FORMAT (1H,24X,5H RLD= ,E10.4,1X,5H RHI= ,E10.4,1X,5HCUE= ,E10.4,AGA02680
+ ,E10.4,1X,3HA= ,E10.4,1X,3HB= ,E10.4/) AGA02690
5 FORMAT (1H,24X,5H RLD= ,E10.4,1X,5H RHI= ,E10.4,1X,5HCUE= ,E10.4,AGA02700
+ ,E10.4,1X,3HA= ,E10.4,1X,4HVIS= ,E10.4/) AGA02710
6 FORMAT (1H,24X,5H RLD= ,E10.4,1X,5H RHI= ,E10.4,1X,4HRC= ,E10.4,1X,AGA02720
+ ,5HALF= ,E10.4,1X,5HGAM= ,E10.4/) AGA02730
7 FORMAT (/1H,5(26H RADIUS RELATIVE NO. >>>)) AGA02740
8 FORMAT (/46H NORMALIZATION FACTOR FOR SIZE DISTRIBUTION = ,E14.7) AGA02850
11 FORMAT (1X,10(1PE12.6,1X))
25 FORMAT (1X,7HN(A) = ,E12.6,2X,9H RBARA = ,E12.6,2X,12H SIGMA(A) =AGA02880
+ ,E12.6,/,1X,7HN(C) = ,E12.6,2X,9H RBARC = ,E12.6,2X,
+ ,12H SIGMA(C) = ,E12.6/) AGA02900
27 FORMAT (1X,42H MARSHALL-PALMER RAIN MODEL : RAIN RATE = ,1PE10.3,AGA02910
+ ,21H MM PER HOUR, DENS = ,1PE12.6,8H PART/CC) AGA02920
45 FORMAT (1H,50H*** BIMODAL DISTRIBUTION...EQUIVALENT DENSITY ISAGA02930
+ ,1PE13.6,18H PARTICLES PER CC/) AGA02940
51 FORMAT (1X,45H AVERAGE ANALYTIC DRY VOLUME PER PARTICLE IS ,3X,AGA02950
+ ,1PE12.6,18H CUBIC MICROMETERS) AGA02960
53 FORMAT (1X,47H AVERAGE NUMERICAL DRY VOLUME IS
+ ,1PE12.6,18H CUBIC MICROMETERS/) AGA02980
54 FORMAT (1X,10X,35H SIZE-INTERVALS USED ARE AS FOLLOWS/)
55 FORMAT (1H,14H INTERVAL NO. ,I3,5X,7HRMIN = ,F11.5,5X,8H RMAX =AGA02990
+ ,F11.5) AGA03000
55 FORMAT (1H,14H INTERVAL NO. ,I3,5X,7HRMIN = ,F11.5,5X,8H RMAX =AGA03010
+ ,F11.5) AGA03020
55 FORMAT (1H,14H INTERVAL NO. ,I3,5X,7HRMIN = ,F11.5,5X,8H RMAX =AGA03030
+ ,F11.5) AGA03030
      RETURN
END

```

```

SUBROUTINE AGXP2( RELHUM, CTSUM, CSSUM, CRSUM, TVOL, TMASS, DENS, CATTN,
1 TEMP, DELTA, NINDX, IW, OLSTAR, OM2, LLLL, IT, WAVE, EM, CAY, EMM, MRTTE, AGB00010
2 PFNZRO) AGB00020
REAL KEXT, KEXTT, KEXOLD AGB00030
N.B. FFF IS AN ALIS FOR ARRAY FF AGB00040
COMMON /PT1/ F(513), R(513), DR(8), RR(9), FFF(514), AGB00050
+NRADI, PI, IDSTP, NKG, NHALV, NI AGB00060
COMMON /PT2/ PHH(65), PSUMTT(65), PGG(65), PSUM(65), PSUMT(65), AGB00070
1PK(65) AGB00080
COMMON /IO/ IGIN, IOUT, NUNIT, IEO, NEOU AGB00090
COMMON /AGXM/ CC(65), W(65), OLT(65), JDIMCK(3) AGB00100
IN THIS SUBROUTINE THE FOLLOWING CONVENTIONS ARE USED IN AGB00110
PREFIXING VARIABLE NAMES: AGB00120
THE LETTER C IS USED FOR CROSS-SECTIONS AGB00130
THE LETTER W IS USED FOR EFFICIENCY FACTORS AGB00140
THE LETTER K IS USED FOR EXTINCTION COEF. PER UNIT PATH (KM) AGB00150
THE LETTER T IS A SUFFIX FOR TOTAL VALUES AGB00160
THE LETTER O IS A PREFIX FOR OMEGA SUB 1 AND 2 CALCULATIONS AGB00170
FOR THE IDSTP=6 AND 12 CASES, DENS IS USED TO TRANSFER THE AGB00180
LIQUID WATER CONTENT FROM THE MAIN PROGRAM TO THIS SUBROUTINE: AGB00190
ELWC IS USED AS THE AEROSOL CONCENTRATION FOR THOSE CASES. AGB00200
IF ( IDSTP.EQ.6.OR.IDSTP.EQ.12 ) ELWC=DENS AGB00210
PZRSMT=0 AGB00220
OLSTAR=0.E0 AGB00230
OM2=0.E0 AGB00240
CTSUMT=0.0E+00 AGB00250
CSSUMT=0.0E+00 AGB00260
DENST=0.E0 AGB00270
CRSUMT=0.0E+00 AGB00280
EMM=1.E0 AGB00290
NLINES=0 AGB00300
BH=1.056E-3 AGB00310
FACTORS BH AND CH ARE USED IN SIZE ADJUSTMENTS AGB00320
FH IS THE SATURATION RATIO AGB00330
FH=RElhum/100.E0 AGB00340
CH=Fh/(1.E0-FH) AGB00350
CONCT=0.0E0 AGB00360
KEXTT=0.E0 AGB00370
CONVERT VOL PER PARTICLE RECEIVED FROM MAIN PROGRAM VIA VARIABLE AGB00380
TVOL TO DRY VOLUME PER PARTICLE IN CUBIC CENTIMETERS AGB00390
DRYVOL=TVOL*1.0E-12 AGB00400
TVOL=0.E0 AGB00410
TMASS=0.E0 AGB00420
DO 6 J=1,IT AGB00430
PSUMTT(J)=0.0E0 AGB00440
PHH(J)=0.0E0 AGB00450
PGG(J)=0.E0 AGB00460
6 CONVERT TEMP. TO KELVIN FOR SUBROUTINE WATER USAGE AGB00470
TEMK=TEMP+273.16E0 AGB00480
SKIP SUBROUTINE WATER FOR THE IDSTP = 12 CASE, AND READ THE AGB00490
OPTICAL DATA FOR RAIN AS EMWA,CAYA,ETC.,.NEEDED BECAUSE CASE AGB00500
IDSTP=12 MAY BE AT WAVELENGTHS LONGER THAN FOUND IN ROUTINE AGB00510
WATER. AGB00520
IF ( IDSTP.EQ.12 ) GO TO 8 AGB00530
SUBROUTINE WATER RETURNS INTERPOLATED VALUES FOR EMW, CAYW AND AGB00540
RHOW AT WAVELENGTH = WAVE AND AT TEMPERATURE = TEMK (DEG K). AGB00550
EMW IS REAL PART OF INDEX OF REFR FOR PURE WATER AT TEMP(DEG C). AGB00560
CAYW IS IMAG. PART OF INDEX OF REFR. FOR PURE WATER. AGB00570
CAYW, HERE IS POSITIVE, BUT TREATED AS NEGATIVE IN MIE-ROUTINE. AGB00580
RHOW IS MASS DENSITY(GM/CC) AT TEMPERATURE = TEMP (DEG C). AGB00590
CALL WATER(WAVE,EMW,CAYW,TEMK,RHOW) AGB00600
WRITE (IOUT,9) EMW,CAYW,TEMP,RHOW AGB00610
BEGIN LOOP OVER AEROSOL COMPONENTS INDEXED BY NK AGB00620
8 DO 32 NK=1,NINDX AGB00630
BYPASS READ OF EMWA,CAYA,ETC. FOR IDSTP=6 CASE..USE WATER DATA AGB00640
IF ( IDSTP.NE.6 ) GO TO 10 AGB00650
EMA=EMW AGB00660
CAYA=CAYW AGB00670
RHWA=RHOW AGB00680
CONC=ELWC AGB00690

```

```

      EMUA=0.0E0          AGB00710
      GO TO 11           AGB00720
C     *** READ OPTICAL AND PHYSICAL DATA ***
10    READ (I0IN,2) EMA,CAYA,EMUA,RHOA,CONC   AGL00730
      IF (IW.EQ.0) EMA=EMU          AGB00740
      IF (IW.EQ.0) CAYA=CAYW        AGB00750
      IF (IDSTP.NE.12) GO TO 11    AGB00760
      EMUA=0.0E0          AGB00770
      RHOW=RHOA            AGB00780
      EMU=EMA             AGB00790
      CAYW=CAYA           AGB00800
11    IF (RHOU.LE.0,E0) RHOA=1.E0          AGB00810
      WRITE (IOUT,3) NK,EMA,CAYA,RHOA,EMUA,CONC  AGB00820
      IF (EMA.LT.1.E-30) GO TO 44    AGB00830
      BHT=BH*(298.E0/TEMK)         AGB00840
      IF (EMUA.LE.0.01) CH=0.0       AGB00850
      BC=BHT*CH                AGB00860
      A=1.E0+((RHOA/RHOW)*EMUA*CH)  AGB00870
      AC=A**((1.E0/3.E0))         AGB00880
C     ADJUST EM, RHO AND CAY PER G. HANEL/ADVANCES IN GEOPHYS/1976  AGB00890
      RHO=RHOW+(RHOA-RHOW)/A      AGB00900
      EM=EMU+(EMA-EMU)/A         AGB00910
      CAY=CAYW+(CAYA-CAYW)/A     AGB00920
      CAY=CAY/EM                AGB00930
C     INITIALIZE QUANTITIES USED TO HOLD RUNNING SUMMATIONS OVER  AGB00940
C     RADII FOR THE CURRENT COMPONENT                         AGB00950
      CTSUM=0.E0              AGB00960
      CSUM=0.E0               AGB00970
      CRSUM=0.E0              AGB00980
      VOL=0.0E0              AGB01000
      DL1SUM=0.0E0            AGB01010
      DL2SUM=0.E0              AGB01020
      PZRSUM=0.               AGB01030
      DO 13 J=1,IT           AGB01040
13    PSUM(J)=0.0E0          AGB01050
      PRINT HEADER IF DETAILED MIE RESULTS ARE TO BE PRINTED  AGB01060
      IF (MORTE.EQ.12345) WRITE (IOUT,5)          AGB01070
      BEGIN ACTUAL LOOP OVER RADIUS INTERVALS FOR THE CURRENT NK VALUE  AGB01080
      THIS LOOP IS THE ONE IN WHICH THE MIE CALCULATIONS ARE CALLED  AGB01090
      INTERVALS ARE INDEXED BY I. THERE ARE NI SUCH INTERVALS.  AGB01100
      DO 26 I=1,NI           AGB01110
      NRADI=2                AGB01120
      D=RR(I+1)-RR(I)        AGB01130
      RIT IS THE ADJUSTED RADIUS FOR THE RELATIVE HUMIDITY TO BE USED  AGB01140
      IN THIS PARTICULAR RUN OR PASS          AGB01150
      RIT=RR(I)*AC-(BC/AC)        AGB01160
      IF (RIT.LT.RR(I).OR.RR(I).LT.0.04E0) RIT=RR(I)  AGB01170
      ALPHA=2.E0*PI*RIT/WAVE    AGB01180
      ROUTINE MIEGX DOES THE ACTUAL MIE CALCULATIONS.  AGB01190
      NOTE THAT THE IMAG. PART OF THE REFRACTIVE INDEX (CAY) HAS BEEN  AGB01200
      NORMALIZED THROUGH DIVISION BY THE REAL PART (EM) BEFORE ITS  AGB01210
      VALUE IS PASSED TO THE MIE-ROUTINE.  AGB01220
      MIEGX RETURNS THE EXTINCTION EFFICIENCY FACTOR AS QT  AGB01230
      MIEGX RETURNS THE SCATTERING EFFICIENCY FACTOR AS QS  AGB01240
      MIEGX RETURNS THE BACK-SCATTERING (RADAR) EFFIC. FACTOR AS QR  AGB01250
      MIEGX RETURNS THE AVERAGE INTENSITY (I1+I2)/2 IN THE ARRAY P()  AGB01260
      AT ANGLES = ARCCOS( C() ), WHERE C() IS SET-UP BY  AGB01270
      SUBROUTINE GUSET OR ANGLE  AGB01280
      MIEGX ALSO RETURNS THE 2-ND AND 3-RD LEGENDRE EXPANSION COEF.  AGB01290
      (OMEGA SUB 1 AND OMEGA SUB 2) AS O1STAR AND O2STAR.  AGB01300
      EMD= (EM)              AGB01310
      CAYD= (CAY)             AGB01320
      ALPHAD= (ALPHA)         AGB01330
      CALL MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,O1STRD,O2STRD,  AGB01340
      +C,IT,PFNZRO)          AGB01350
      EM=(EMD)                AGB01360
      CAY=(CAYD)              AGB01370
      ALPHA=(ALPHAD)          AGB01380
      QT=(QTD)                AGB01390
      QS=(QSD)                AGB01400

```

```

QR=(QRD)
01STAR=(01STRD)
02STAR=(02STRD)
KK=1+(I-1)*NKG
IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I),F(KK),ALPHA,QT,QS,QR AGB01410
FKK=F(KK)
FKKA=FKK*PI*RIT**2.E0 AGB01420
VOLHH=4.1888*FKK*RIT**3.E0 AGB01430
OL1HH=01STAR*FKKA*QT AGB01440
OL2HH=02STAR*FKKA*QT AGB01450
CTHH=QT*FKKA AGB01460
CSHH=QS*FKKA AGB01470
CRHH=QR*FKKA AGB01480
DO 14 J=1,IT AGB01490
PHH(J)=P(J)*FKK AGB01500
CONTINUE AGB01510
14 PFNZER=PFNZR0*FKK AGB01520
RIT=RR(I+1)*AC-(BC/AC) AGB01530
IF (RIT.LT.RR(I+1).OR.RR(I+1).LT.0.04E0) RIT=RR(I+1) AGB01540
ALPHA=2.E0*PI*RIT/WAVE AGB01550
EMD=(EM) AGB01560
CAYD=(CAY) AGB01570
ALPHAD=(ALPHA) AGB01580
CALL MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,01STRD,02STRD, AGB01590
+C,IT,PFNZR0) AGB01600
EM=(EMD) AGB01610
CAY=(CAYD) AGB01620
ALPHA=(ALPHAD) AGB01630
QT=(QTD) AGB01640
QS=(QSD) AGB01650
QR=(QRD) AGB01660
01STAR=(01STRD) AGB01670
02STAR=(02STRD) AGB01680
KK1=1+NKG*I AGB01690
FKK1=F(KK1) AGB01700
IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I+1),FKK1,ALPHA,QT,QS,QR AGB01710
FKK1A=FKK1*PI*RIT**2.E0 AGB01720
VOLHH=(VOLHH+4.188E0*FKK1*RIT**3.E0)*D*0.5E0 AGB01730
OL1HH=(OL1HH+FKK1A*QT*01STAR)*D*0.5E0 AGB01740
OL2HH=(OL2HH+FKK1A*QT*02STAR)*D*0.5E0 AGB01750
CTHH=(CTHH+QT*FKK1A)*D*.5E0 AGB01760
CSHH=(CSHH+QS*FKK1A)*D*.5E0 AGB01770
CRHH=(CRHH+QR*FKK1A)*D*.5E0 AGB01780
DO 15 J=1,IT AGB01790
PHH(J)=(PHH(J)+P(J)*FKK1)*D*0.5E0 AGB01800
CONTINUE AGB01810
15 PFNZER=(PFNZER+PFNZR0*FKK1)*D*0.5 AGB01820
FF=0.5E0*D*(FKK+FKK1) AGB01830
NT=1 AGB01840
N=1 AGB01850
16 NJ=NT AGB01860
NT=2*NT AGB01870
D=0.5E0*D AGB01880
VOLGG=0.0E0 AGB01890
CL1GG=0.0E0 AGB01900
OL2GG=0.E0 AGB01910
CTGG=0.E0 AGB01920
CSGG=0.E0 AGB01930
CRGG=0.E0 AGB01940
FT=0.E0 AGB01950
DO 17 J=1,IT AGB01960
17 PGG(J)=0.E0 AGB01970
PZRTMP=0. AGB01980
C NEXT LOOP HANDLES INTERMEDIATE PARTICLE SIZES.. THOSE LYING BETWEEN AGB02040
RMIN AND RMAX FOR THE CURRENT INTERVAL WHOSE INDEX IS I. AGB02050
DO 19 JC=1,NJ AGB02060
KK=1+(I-1)*NKG+(2*JG-1)*(NKG/NT) AGB02070
RIT=R(KK)*AC -(BC/AC) AGB02080
IF (RIT.LT.R(KK).OR.R(KK).LT.0.04E0) RIT=R(KK) AGB02090
ALPHA=2.E0*PI*RIT/WAVE AGB02100

```

```

EMD= (EM)
CAYD= (CAY)
ALPHAD= (ALPHA)
CALL MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,01STRD,02STRD,
+C, IT,PFNZRO)
EM=(EMD)
CAY=(CAYD)
ALPHA=(ALPHAD)
QT=(QTD)
QS=(QSD)
QR=(QRD)
01STAR=(01STRD)
02STAR=(02STRD)
IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,R(KK),F(KK),ALPHA,QT,QS,QR
NRADI=NRADI+1
FKK=F(KK)
FKKA=FKK*PI*RIT**2.E0
VOLGG=4.1888E0*FKK*RIT**3.0E0+VOLGG
DL1GG=DL1GG+01STAR*FKKA*QT
DL2GG=DL2GG+02STAR*FKKA*QT
CTGG=CTGG+QT*FKKA
CSGG=CSGG+QS*FKKA
CRGG=CRGG+QR*FKKA
DO 18 J=1, IT
PGG(J)=PGG(J)+P(J)*FKK
18 CONTINUE
PZRTMP=PZRTMP+PFNZRO*FKK
19 FT=FT+FKK
C ADD RESULTS ACCUMULATED DURING PREVIOUS HALVINGS TO THOSE FOUND
C FOR THE NEW RADII TREATED WITHIN THE LOOP OVER INDEX JG
VOLHHT=0.5E0*VOLHH+D*VOLGG
DL1HHT=0.5E0*DL1HH+D*DL1GG
DL2HHT=0.5E0*DL2HH+D*DL2GG
CTHHT=0.5E0*CTHH+D*CTGG
CSHH=0.5E0*CSHH+D*CSGG
CRHH=0.5E0*CRHH+D*CRGG
DO 20 J=1, IT
20 PHH(J)=.5E0*PHH(J)+D*PGG(J)
PFNZER=.5E0*PFNZER+D*PZRTMP
FFT=0.5E0*FF+D*FT
IF (CTHHT.LT.1.E-30) GO TO 22
DEL=ABS(VOLHHT-VOLHH)/ABS(VOLHHT)
IF (DEL.LE.DELTA) GO TO 21
GO TO 22
21 IF (N.GT.2) GO TO 24
DO NOT ALLOW DEL LESS THAN DELTA EXIT UNLESS AT LEAST TWO
HAVINGS HAVE BEEN DONE
22 IF (N.EQ.NHALV) GO TO 24
C MUST EXIT WHEN NHALV HALVINGS HAVE BEEN DONE EVEN IF THE DELTA
C CRITERION HAS NOT BEEN SATISFIED..SINCE NO MORE VALUES OF RADII
C ARE AVAILABLE.
FF=FFT
CRHH=CRHH
DL1HH=DL1HHT
DL2HH=DL2HHT
CSHH=CSHH
CTHH=CTHH
VOLHH=VOLHHT
N=N+1
GO TO 16
24 CONTINUE
IF(N.EQ.NHALV) WRITE(IOUT,124) I
C SUM QUANTITIES OVER ALL INTERVALS TREATED UP UNTIL NOW
CTSUM=CTSUM+CTHHT
CSSUM=CSSUM+CSHH
CRSUM=CRSUM+CRHH
VOL=VOL+VOLHHT
OL1SUM=OL1SUM+OL1HHT
OL2SUM=OL2SUM+OL2HHT
C AT THIS POINT, PSUM<> IS THE RUNNING SUM OF THE AVG. INTENSITY

```

```

C AS SUMMED OVER SIZES
DO 25 J=1, IT
25 PSUM(J)=PSUM(J)+PHH(J)
PZRSUM=PZRSUM+PFNZER
WRITE (IOUT,27) NK,I,NRADI,CTHHT
NLINES=NLINES+NRADI
C END LOOP OVER HALVING INTERVALS INDEXED BY I
26 CONTINUE
C CALCULATE PARTICLE NUMBER DENSITY( NO. PER CC ) AS DENSC
DENSC=CONC/(RHOA*DRYVOL)
C OVERRIDE CALCULATED VALUE OF DENSC WITH DENS IF LLLL =1
IF (LLL, EQ, 1) DENSC=DENS
C RECALCULATE CONC FROM OTHER INPUT DATA IF LLLL=1
IF (LLL, EQ, 1) CONC=DENS*RHOA*DRYVOL
C REPLACE DENS BY DENSC FOR LATER USE BY AGXP3
DENS=DENSC
C WEIGHT CTSUM, ETC. BY NUMBER DENSITIES (DENSC) FOR THIS COMPONENT
CTSUM=CTSUM*DENS
CSSUM=CSSUM*DENS
CRSUM=CRSUM*DENS
VOL=VOL*DENS
OL1SUM=OL1SUM*DENS
OL2SUM=OL2SUM*DENS
DO 29 J=1, IT
29 PSUM(J)=PSUM(J)*DENSC
PZRSUM=PZRSUM*DENS
C NOW, SUM OVER COMPONENTS INDEXED BY NK
CONCT IS THE TOTAL DRY-AEROSOL CONCENTRATION IN MG PER CC
CONCT=CONCT+1.E3*CONC
DENST=DENST+DENSC
OLSTAR=OL1SUM+OLSTAR
OM2=OL2SUM+OM2
CTSUMT=CTSUMT+CTSUM
AT THIS POINT, CTSUMT IS THE TOTAL EXTINCTION CROSS SECTION
(IN SQ. MICRONS) AS SUMMED OVER ALL COMPONENTS WHICH
HAVE BEEN DEALT WITH THUS FAR
CSSUMT=CSSUMT+CSSUM
CRSUMT=CRSUMT+CRSUM
DO 30 J=1, IT
30 PSUMTT(J)=PSUM(J)+PSUMTT(J)
PZRSMT=PZRSMT+PZRSUM
VOL=VOL*1.E-12
TVOL IS THE TOTAL VOLUME (IN CM**3) OCCUPIED BY THE AEROSOL
PARTICLES; TVOL IS NOT ACTUALLY USED IN THIS VERSION OF
PROGRAM AGAIN.
TVOL=VOL+TVOL
EMASS=VOL*RHO
TMASS=TMASS+EMASS
KEXTOLD=KEXTT
KEXTT=CTSUMT*1.E-3
KEXT=KEXTT-KEXTOLD
WRITE (IOUT,31) NK,VOL,EMASS,KEXT
VFC(VOL) IS THE VOLUME PACKING FRACTION; THAT IS, THE FRACTION
OF EACH CC OF SPACE WHICH IS FILLED BY AEROSOL MATERIAL BELONGING
TO THE CURRENT COMPONENT NK.
TMASS IS THE TOTAL MASS OF AEROSOL FOUND IN 1 CC OF SPACE.
EMASS IS THE MASS OF AEROSOL MATERIAL PER CC ASSOCIATED WITH
THE CURRENT COMPONENT NK.
KEXT IS THE EXTINCTION COEF.(PER KM) WHICH IS ASSOCIATED WITH
THE CURRENT COMPONENT--AS IF IT ALONE WERE PRESENT.
KEXTT IS THE SUM OF THE KEXT'S OVER ALL COMPONENTS.
END LOOP OVER AEROSOL COMPONENTS INDEXED BY NK.
IF (NINDX.GT.1) WRITE (IOUT,42) NK
IF (MQRT.EQ.12345) WRITE (IOUT,43)
32 CONTINUE
IF (NINDX.GT.1) WRITE (IOUT,33) TMASS,KEXTT
WRITE (IOUT,34) NLINES
DENS=DENST
NRADI=NLINES
C NOW, PERFORM THE FINAL RENORMALIZATIONS TO OBTAIN CTSUM, ETC.

```

```

C VALUES REPRESENTATIVE OF A SINGLE AVERAGE PARTICLE.
C CTSUM BECOMES THE EXTINCTION CROSSSECTION IN SQ. MICROMETERS PER
C AVERAGE PARTICLE. THE OTHER QUANTITIES CARRY SIMILAR MEANINGS.
C DO 35 J=1, IT
35 PSUM(J)=PSUMT(J)/DENST
    OLSTAR=OLSTAR/CTSUM
    OM2=OM2/CTSUM
    CTSUM=CTSUM/DENST
    CSSUM=CSSUM/DENST
    CRSUM=CRSUM/DENST
    .AND CONVERT ANG INTENS AT ZERO DEGREES INTO PHASE FUNCT.
    WITH NORMALIZATION
    PFNZRO=PZRSMT/DENST*(WAVE*WAVE/(PI*CTSUM*EMM*EMM))
    WRITE(IOUT,36) OLSTAR,OM2,PFNZRO
C CALCULATE ATTENUATION COEFS. IN SQ.METERS PER MILLIGRAM
    CATTN=CTSUM*1.E-12/CONCT
    CATTNW=1.E-12*KEXT/TMASS
    WRITE(IOUT,37) CATTN
    IF(CRELHUM.GT.1.0)WRITE(IOUT,38) CATTNW
    GO TO 41
44 WRITE(IOUT,45) EMA
    STOP
2 FORMAT(4F10.6,E15.7)
3 FORMAT(1H,6H INDEX=,I3,4H M= ,F10.6,6H K = -,F10.6,9HI. MASS ,
+ 10HDENSITY = ,F8.6,
+ 17H GROWTH FACTOR = ,F8.4,9H. CONC = ,1PE12.5,7H GM/CC)
4 FORMAT(1X,F10.5,6(2X,1PE11.5))
5 FORMAT(//54H R(MICRONS) DRY RADIUS N(R) MIE SIZE
+ 36HQ (EXT) Q(SCA) Q(RADAR))
9 FORMAT(1H,39HINDEX OF REFRACTION FOR PURE WATER IS: ,F8.6,
+ 3H -,F8.6,1HI//,1X,25HMASS DENSITY OF WATER AT ,F6.2,
+ 11H DEG C IS: ,F8.2,6H GM/CC,/)
124 FORMAT(/52H *** CONVERGENCE LEVEL NOT REACHED FOR INTERVAL NO. ,
+ 13,4H ***)
27 FORMAT(1H,19H FOR COMPONENT NO. ,I3,15H INTERVAL NO. ,I3,1H ,I4,
+ 43H RADII WERE USED. CONTRIBUTION TO CTSUM = ,1PE12.6)
31 FORMAT(1H //,20H FOR COMPONENT NO. ,I2,12H VPF = ,1PE12.5,
+ 7H PER CC,24H MASS CONCENTRATION = ,E12.5,21H GM/CC. KEXT =
+ E12.5,7H PER KM)
33 FORMAT(1H //,29H TOTAL MASS CONCENTRATION = ,1PE12.5,7H GM/CC),
+ 15H TOTAL KEXT = ,E12.5,7H PER KM)
34 FORMAT(/1X,32H TOTAL NUMBER OF RADII USED WAS ,I5)
36 FORMAT(/1X,19H ANALYTIC SOLUTIONS/,
+ 16H OMEGA SUB 1 = ,1PE14.7/,16H OMEGA SUB 2 = ,1PE14.7/
+ ,16H PFN AT ZERO = ,1PE14.7/)
37 FORMAT(1H,21H ATTENUATION COEF. = ,1PE12.5,12H SQ-METERS/,
+ 33HMILLIGRAM OF DRY AEROSOL MATERIAL)
38 FORMAT(1H,21H ATTENUATION COEF. = ,1PE12.5,12H SQ-METERS/,
+ 33HMILLIGRAM OF WET AEROSOL MATERIAL)
42 FORMAT(1H //,1X,10X,30(1H*),33H END OF AEROSOL COMPONENT CYCLE ,
+ 7HNNUMBER ,I3,2X,30(1H*)//)
43 FORMAT(1H )
45 FORMAT(///,1X,11H***** EMA (,F10.6,20H) IS EITHER ZERO OR ,
+ 35HNEGATIVE - PROGRAM TERMINATED *****)
41 RETURN
    END

```

```

SUBROUTINE AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,
+WAVE,EM,CAY,EMM,IT,IEND,IANG) AGC00010
COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65) AGC00020
COMMON /IO/ IJOIN,IOUT,NUNIT,IEO,NEOU AGC00030
COMMON /AGXM/ C(65),WC(65),OLT(65),JDIMCK(3) AGC00040
COMMON /PT1/ F(513),RK(513),DR(8),RR(9),FF(514) AGC00050
+ ,NRADI,PI, IDSTP,NKG,NHALV,NI AGC00060
C IEND=1 WHEN THE COMPOSITE PHASE FUNCTION IS BEING WRITTEN AGC00070
IF (IEND.EQ.1) GO TO 6 AGC00080
ALBDO=CSSUM/CTSUM AGC00090
C PFACT IS USED TO CONVERT AVG. INTENSITY PSUM() INTO PHASE- AGC00110
FUNCTIONS. SFACT IS USED TO CONVERT PSUM INTO SCATTERING AGC00120
FRACTIONS, NORMALIZED PER SUM. THE INTEGRAL OF SCAT OVER SOLID AGC00130
ANGLE SHOULD YIELD THE TOTAL SCATTERING CROSS-SECTION IN SQ. M. AGC00140
SFACT=WAVE*WAVE*DENS*1.E-6/(4.*PI*PI) AGC00150
PFACT=WAVE*WAVE/(PI*CTSUM*EMM*EMM) AGC00160
DO 15 J=1,IT AGC00170
SCAT=PSUM(J)*SFACT AGC00180
PSUM(J)=PSUM(J)*PFACT AGC00190
C UNCOMMENT THE FOLLOWING STMT IF U WANT SCAT FRACT,COSINES AND A AGC00200
COUNTER WRITTEN ON NUNIT. NUNIT IS SET IN THE BLOCK DATA ROUTINE. AGC00210
WRITE (NUNIT,3) SCAT,C(J),J AGC00220
15 CONTINUE AGC00230
IF (NINDX.GE.2) WRITE (IOUT,12) AGC00240
WRITE (IOUT,4) IDSTP,WAVE,EM,CAY,CTSUM,CSSUM,ALBDO AGC00250
C CONVERT AVG. CROSS-SECTIONS TO COEFFICIENTS (PER KM) AGC00260
CTSUM=CTSUM*1.0E-3*DENS AGC00270
CSSUM=CSSUM*1.0E-3*DENS AGC00280
CRSUM=CRSUM*1.0E-3*DENS AGC00290
WRITE (IOUT,13) CTSUM,CSSUM,CRSUM AGC00300
WRITE (IOUT,14) GNU,DENS AGC00310
IF (IT.LT.2) GO TO 21 AGC00320
C WRITE PHASE FUNCTION AT SPECIFIED ANGLES AGC00330
6 WRITE (IOUT,1) AGC00340
WRITE (IOUT,5) AGC00350
C FIND ANGLES FROM COSINES AGC00360
DO 2 I=1,IT AGC00370
2 FF(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI AGC00380
DO 19 J=1,IT,4 AGC00390
K=J+3 AGC00400
IF (K.GT.IT) K=IT AGC00410
19 WRITE (IOUT,8) (C(I),FF(I),PSUM(I),I=J,K) AGC00420
IF (IANG.NE.0) RETURN AGC00430
C ROUTINE GAUS GENERATES AND PRINTS THE LEGENDRE AGC00440
EXPANSION COEFS (OMEGAS) FOR THE PHASE FUNCTION. AGC00450
CALL GAUS(IT) AGC00460
C CHECK TO SEE IF SNG. SCAT. ALBEDO (ALBDO) COMPUTED DIRECTLY AGC00470
FROM CROSS-SECTIONS AGREES WITH THAT FOUND FROM THE LEGENDRE AGC00480
EXPANSION OF PHASE-FUNCTION. AGC00490
IF ((ABS(OL(1)-ALBDO)/ALBDO.GT.5.E-3).AND.(IEND.NE.1)) AGC00500
1 WRITE (IOUT,20) AGC00510
1 FORMAT (//1H ,50X,14HPHASE FUNCTION/,1X,42X,31H(NORMALIZED TO 4 PI)AGC00520
+ OMEGA ZERO)//) AGC00530
3 FORMAT (2(E13.7,1X),I3) AGC00540
4 FORMAT (1H1//,41H DISTRIBUTION WAVELENGTH REFRACTIVE,9X, AGC00550
+ 20HEXTINCTION X SECTION,8X,20HSCATTERING X SECTION,12X,5HALBDO/, AGC00560
+ 1H ,6X, 4HTYPE,6X,9H(MICRONS),8X,5HINDEX,16X,12H(SQ MICRONS),13X, AGC00570
+ 12H(SQ MICRONS)/1H ,I9,4X,F11.4,F10.4,3H(1-,F7.4,2H), AGC00580
+ 7X,1PE14.7,11X,1PE14.7,12X,1PE14.7/) AGC00590
5 FORMAT (1H ,3X,4(5H MU,2X, ANGLE ,17H PHASE FUNCTION )) AGC00600
8 FORMAT (1H ,F9.5,F7.2,E12.5,3(3X,F9.5,F7.2,E12.5)) AGC00610
12 FORMAT (52H THIS IS A MIXED CASE * SUBSEQUENT REFRACTIVE INDEX , AGC00620
+ 34PRINT-OUTS ARE NOT GENERALLY VALID/) AGC00630
13 FORMAT (1H ,10H K(EXT) = ,1PE13.7,11H; K(SCA) = ,E13.7, AGC00640
+ 11H; K(RAD) = ,E13.7,11H ALL PER KM//) AGC00650
14 FORMAT (//1H WAVENUMBER = ,1PE12.6,5H CM-1,5X,10HDENSITY = ,E12.6,AGC00660
+ 17H PARTICLES PER CC//) AGC00670
20 FORMAT (//12H *** VALUES AGC00680
+ ,55HOF ALBDO AND OL(1) DISAGREE BY MORE THAN 0.5 PERCENT **AGC00690
+ ,34HLARGER VALUE OF 'IT' IS NEEDED **//) AGC00700

```

21

RETURN
END

AGC00710
AGC00720

```

C SUBROUTINE ANGLE(PI,IANG,IT)
CC THIS ROUTINE IS TO BE USED TO REPLACE GUSET FOR THE PURPOSE
CC OF USING AGAUS TO DO PHASE FUNCTION CALCULATIONS AT -IT-
CC ANGLES BETWEEN 0 AND 180 DEGREES, RATHER THAN AT THE Q-L
CC QUADRATURE ABSCISSA VALUES. IT ALSO READS THE INPUT ANGLES
CC IF IANG=2.
COMMON /I0/I0IN,I0UT,NUNIT,IE0,NEOU
COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
RADS=PI/180.
DEL=180./FLOAT(IT-1)
IF (IE0.EQ.1.OR.IE0.EQ.2.OR.IE0.EQ.5) GO TO 4
IF (IANG.EQ.2) GO TO 2
DO 1 I=1,IT
1 W(I)=DEL*FLOAT(I-1)
2 C(I)=COS(W(I))*RADS
3 RETURN
4 READ(5,100) (W(I),I=1,IT)
5 DO 3 I=1,IT
3 C(I)=COS(W(I))*RADS
6 IANG=1
7 RETURN
100 FORMAT(16F5.1)
END

```

```

ANG00010
ANG00020
ANG00030
ANG00040
ANG00050
ANG00060
ANG00070
ANG00080
ANG00090
ANG00100
ANG00110
ANG00120
ANG00130
ANG00140
ANG00150
ANG00160
ANG00170
ANG00180
ANG00190
ANG00200
ANG00210
ANG00220
ANG00230

```

```

C FUNCTION GAMMA(X) GMA00010
C GAMMA FUNCTION: TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS, GMA00020
C ABRAMOWITZ AND STEGUN, NOV 1964, PP 256-257. RECURRENCE FORMULA GMF00030
C 6.1.16, POLYNOMIAL APPROXIMATION 6.1.35. GMA00040
C DATA A1,A2,A3,A4,A5 GMA00050
C + .-5748646, .9512363,-.6998588, .4245549,-.1010678/ GMA00060
C COMPUTER AND GAMMA FUNCTION LIMITS GMA00070
C IF (X.GT.34.,OR.X.LT.0.) GO TO 3 GMA00080
C GSUM=1 GMA00090
C N=IFIX(X+.00001) GMA00100
C FIND Z.LE. 1. GMA00110
C Z=X-FLOAT(N) GMA00120
C CK FOR Z BEING INTEGER GMA00130
C IF (Z.LT.1.E-04) N=N-1 GMA00140
C IF (Z.LT.1.E-04) Z=1 GMA00150
C COMPENSATE FOR N-1 IN FORMULA GMA00160
C N=N-1 GMA00170
C IF Z.LE. 1. SKIP LOOP GMA00180
C IF (N.LE.0) GO TO 2 GMA00190
C RECURRENCE RELATION: G(N+Z)=(N-1+Z)*(N-2+Z)...(1+Z)*G(1+Z) GMA00200
C DO 1 I=1,N GMA00210
C VALUE=FLOAT(I)+Z GMA00220
C GSUM=GSUM*VALUE GMA00230
C POLYNOMIAL APPROXIMATION: Z.LE.1 GMA00240
C GAMMA=1.+A1*Z+A2*Z*Z+A3*Z*Z*Z+A4*Z*Z*Z*Z+A5*Z*Z*Z*Z*Z GMA00250
C GAMMA=GAMMA*GSUM GMA00260
C RETURN GMA00270
C 3 WRITE (1,100) X GMA00280
100 FORMAT(1H '***** THE VALUE OF X (' 2PE11.4,') IS EITHER ', GMA00290
C + 'OUTSIDE COMPUTER LIMITS',/, ' OR NEGATIVE - PGM STOPPED *****') GMA00300
C STOP GMA00310
C END GMA00320

```

```

C SUBROUTINE GUSSET( IT )
C THIS ROUTINE CALCULATES THE ABSCISSAE C()
C AND GAUSS-LEGENDRE WEIGHTS W() FOR NUMERICAL INTEGRATION
C VIA GAUSS-LEGENDRE QUADRATURE OF ORDER N
COMMON /IO/ IOUN,IOUT,NUNIT,IEO,NEOU
COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
+ ,NRADI,PI,IDSTP,NKG,NHALV,NI
N=IT
TOL=1.0E-06
AA=2.0E+00/PI**2.0E+00
AB=-62.0E+00/(3.0E+00*PI**4.0E+00)
AC=15116.0E+00/(15.0E+00*PI**6.0E+00)
AD=-12554474.0E+00/(105.0E+00*PI**8.0E+00)
FC(1)=1.0E+00
EN= FLOAT(N)
NP1=N+1
U=1.0E+00-(2.0E+00/PI)**2.0E+00
D=1.0E+00/SQRT((EN+0.5E+00)**2.0E+00+U/4.0E+00)
DO 1 I=1,N
S= FLOAT(I)
A=4.0E+00*S-1.0E+00
AE=AA/A
AF=AB/A**3.0E+00
AG=AC/A**5.0E+00
AH=AD/A**7.0E+00
1 R(I)=PI*(A+AE+AF+AG+AH)/4.0E+00
DO 6 K=1,N
X=COS(R(K)*D)
2 F(2)=X
DO 3 NN=3,NP1
ENN= FLOAT(NN-1)
F(NN)=((2.0E+00*ENN-1.0E+00)*X*F(NN-1)-(ENN-1.0E+00)*F(NN-2))/ENN
IF (ABS(F(NN)),GT,1E+35) F(NN)=SIGN(1.0E+35,F(NN))
PNP=ENN*(F(N)-X*F(NP1))/(1.0E+00-X*X)
XI=X-F(NP1)/PNP
XD= ABS(XI-X)
XDD=XD-TOL
IF (XDD) 5,5,4
4 X=XI
GO TO 2
5 C(K)=X
W(K)=2.0E+00*(1.0E+00-X*X)/(ENN*F(N)*ENN*F(N))
DO 7 I=1,N
R(I)=0.00
7 F(I)=0.00
RETURN
END

```

GUSSET010
GUSSET020
GUSSET030
GUSSET040
GUSSET050
GUSSET060
GUSSET070
GUSSET080
GUSSET090
GUSSET100
GUSSET110
GUSSET120
GUSSET130
GUSSET140
GUSSET150
GUSSET160
GUSSET170
GUSSET180
GUSSET190
GUSSET200
GUSSET210
GUSSET220
GUSSET230
GUSSET240
GUSSET250
GUSSET260
GUSSET270
GUSSET280
GUSSET290
GUSSET300
GUSSET310
GUSSET320
GUSSET330
GUSSET340
GUSSET350
GUSSET360
GUSSET370
GUSSET380
GUSSET390
GUSSET400
GUSSET410
GUSSET420
GUSSET430
GUSSET440
GUSSET450
GUSSET460
GUSSET470
GUSSET480

```

SUBROUTINE MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,01STRD,02STRD, MIEGX010
+C,IT,PFNZRO) MIEGX020
C THIS ROUTINE IS CURRENTLY SINGLE PRECISION COMPLEX MIEGX030
C CHANGE THE VALUE OF NDIM IF YOU CHANGE THE DIM OF A IN NEXT LINE MIEGX040
C COMPLEX AC(600),ACAPN,ZNUM,ZDEN,ZPDT,ZRPDT,ZAN,ZANF,YRF,RRF, MIEGX050
1 RRFX,WM1,FNA,FNB,TC1,FNAP,FNBP,FNAPP,FNBPP,TC2,WFN(2) MIEGX060
COMMON /IO/ I0IN,I0UT,NUNIT,IE0,NEOU MIEGX070
DIMENSION P(65),C(65) MIEGX080
DIMENSION T(4),TA(4),TB(2),TC(2),TD(2),TE(2),TF(2),TG(2) MIEGX090
DIMENSION ELTRMX(4,76),PI(3,76),TAU(3,76) MIEGX100
EQUIVALENCE (WFN(1),TA(1)),(FNA,TB(1)),(FNB,TC(1)),(FNAP,TD(1)) MIEGX110
EQUIVALENCE (FNBP,TE(1)),(FNAPP,TF(1)),(FNBPP,TG(1)) MIEGX120
C THESE EQUIVALENCES ALLOW USE OF REAL AND IMAG PARTS INDIVIDUALLY MIEGX130
TOL = 1.E-06 MIEGX140
ITT=IT MIEGX150
X=(ALPHAD) MIEGX160
CAY=(CAYD) MIEGX170
EM=(EMD) MIEGX180
CAYE=CAY*EM MIEGX190
GRF=0.0 MIEGX200
S=1.0 MIEGX210
RF=CMPLX(EM,-CAYE) MIEGX220
NMX=IFIX(X*(EM+CAYE))+9 MIEGX230
RRF=1.0/RF MIEGX240
RX=1.0/X MIEGX250
RRFX=RRF*RX MIEGX260
C LOOP POINT FOR CALCULATING PFN AT ZERO DEGREES MIEGX270
IAPXCT=0 MIEGX280
21 CONTINUE MIEGX290
C THESE ARE THE PI AND TAU FUNCTIONS MIEGX300
DO 1 J=1,IT MIEGX310
PI(1,J)=0.0 MIEGX320
PI(2,J)=1.0 MIEGX330
TAU(1,J)=0.0 MIEGX340
TAU(2,J)=C(J) MIEGX350
1 CONTINUE MIEGX360
T(1)=COS(X) MIEGX370
T(2)=SIN(X) MIEGX380
WM1=CMPLX(T(1),-T(2)) MIEGX390
WFN(1)=CMPLX(T(2),T(1)) MIEGX400
WFN(2)=RX*WFN(1)-WM1 MIEGX410
T(1)=CAYE*X MIEGX420
N=1 MIEGX430
C NDIM MUST EQUAL THE DIMENSION OF AC( ). MIEGX440
NDIM=600 MIEGX450
IF (NMX.LT.NDIM) NDELTA=NMX MIEGX460
IF (NMX.GT.NDIM) NDELTA=NDIM MIEGX470
NMX=0 MIEGX480
IF (N.EQ.1) GO TO 4 MIEGX490
2 EN=FLOAT(N) MIEGX500
T(1)=2.0*EN-1.0 MIEGX510
T(2)=EN-1.0 MIEGX520
T(3)=2.0*EN+1.0 MIEGX530
DO 3 J=1,IT MIEGX540
PI1J=PI(1,J) MIEGX550
PI2J=PI(2,J) MIEGX560
CJ=C(J) MIEGX570
C SWITCH FOR CALCULATING PFN AT ZERO DEGREES MIEGX580
IF (IAPXCT.EQ.1) CJ=1.0 MIEGX590
S2T=1.0-CJ*CJ MIEGX600
PI(3,J)=(T(1)*PI2J*CJ-EN*PI1J)/T(2) MIEGX610
TAU(3,J)=CJ*(PI(3,J)-PI1J)-T(1)*S2T*PI2J+TAU(1,J) MIEGX620
3 CONTINUE MIEGX630
WM1=WFN(1) MIEGX640
WFN(1)=WFN(2) MIEGX650
WFN(2)=T(1)*RX*WFN(1)-WM1 MIEGX660
4 CONTINUE MIEGX670
C CALCULATE RATIO OF BESSSEL FNS OF CONSECUTIVE ORDER MIEGX680
IF (N.LT.(NMX+1)) GO TO 9 MIEGX690
NMX=NMX+NDELTA MIEGX700

```

```

NMIN=NMX+1-NDELTA      MIEGX710
V=FLOAT(NMX)+0.50       MIEGX720
Y=RF*X                  MIEGX730
ZANP=2.0/Y                MIEGX740
ZNUM=ZANP*V                MIEGX750
ZPDT=ZNUM                 MIEGX760
V=V+1.0                  MIEGX770
ZDEN=ZANP*V                MIEGX780
ZNUM=ZDEN-1.0/ZNUM        MIEGX790
5 ZRPDT=ZNUM/ZDEN        MIEGX800
ZPDT=ZRPDT*ZPDT          MIEGX810
IF (ABS(REAL(ZPDT))-1.0) LT. TOL GO TO 7 MIEGX820
IF (V.LT.20000.0) GO TO 6 MIEGX830
WRITE (IOUT,1000) X,EM,CAYE MIEGX840
STOP MIEGX850
6 V=V+1.0                  MIEGX860
ZAN=ZANP*V                MIEGX870
ZNUM=ZAN-1.0/ZNUM        MIEGX880
ZDEN=ZAN-1.0/ZDEN        MIEGX890
GO TO 5                  MIEGX900
7 CONTINUE                 MIEGX910
J=NMX                     MIEGX920
8 JJ=J-NMX+NDELTA        MIEGX930
A(JJ)=-(FLOAT(J))/Y+ZPDT MIEGX940
J=J-1                     MIEGX950
IF (J.LT.NMIN) GO TO 9 MIEGX960
ZPDT=(2.0*FLOAT(J)+1.0)/Y-1.0/ZPDT MIEGX970
GO TO 8                  MIEGX980
9 CONTINUE                 MIEGX990
J=N-NMX+NDELTA          MIEGX000
ACAPN=A(J)                MIEGX010
IF (N.GT.1) GO TO 11      MIEGX020
THIS PART FOR N EQUAL 1 ONLY MIEGX030
TC1=ACAPN*RRF+RX          MIEGX040
TC2=ACAPN*RF+RX          MIEGX050
C SEE EQUIVALENCE STMTS FOR EXPANATION OF TAC >, ETC. MIEGX060
FNA=(TC1*TAK(3)-TAK(1))/(TC1*WFN(2)-WFN(1)) MIEGX070
FNB=(TC2*TAK(3)-TAK(1))/(TC2*WFN(2)-WFN(1)) MIEGX080
FNAP=FNA                  MIEGX090
FNBP=FNB                  MIEGX100
T(1)=1.50                 MIEGX110
TB(1)=T(1)*TB(1)          MIEGX120
TB(2)=T(1)*TB(2)          MIEGX130
TC(1)=T(1)*TC(1)          MIEGX140
TC(2)=T(1)*TC(2)          MIEGX150
DO 10 J=1,IT              MIEGX160
TAU2J=TAU(2,J)            MIEGX170
ELTRMX(1,J)=TB(1)+TC(1)*TAU2J MIEGX180
ELTRMX(2,J)=TB(2)+TC(2)*TAU2J MIEGX190
ELTRMX(3,J)=TC(1)+TB(1)*TAU2J MIEGX200
ELTRMX(4,J)=TC(2)+TB(2)*TAU2J MIEGX210
10 CONTINUE                 MIEGX220
QEXT=2.0*(TB(1)+TC(1))    MIEGX230
QSCAT=(TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2))/0.750 MIEGX240
O1STAR=0.0                 MIEGX250
O2STAR=0.0                 MIEGX260
SUMRR=2.0*(TB(1)-TC(1))   MIEGX270
SUMRI=2.0*(TB(2)-TC(2))   MIEGX280
N=2                         MIEGX290
GO TO 2                  MIEGX300
11 CONTINUE                 MIEGX310
TC1=ACAPN*RRF+EN*RX       MIEGX320
TC2=ACAPN*RF+EN*RX       MIEGX330
C SEE EQUIVALENCE STMTS FOR EXPLANATION OF TAC >, ETC. MIEGX340
FNA=(TC1*TAK(3)-TAK(1))/(TC1*WFN(2)-WFN(1)) MIEGX350
FNB=(TC2*TAK(3)-TAK(1))/(TC2*WFN(2)-WFN(1)) MIEGX360
T(4)=T(1)/(EN*T(2))       MIEGX370
T(2)=(TC(2)*(EN+1.0))/EN MIEGX380
S=-S                      MIEGX390
SUMRR=SUMRR+S*T(3)*(TB(1)-TC(1)) MIEGX400

```

```

C   SUMRI=SUMRI+S*T(3)*(TB(2)-TC(2))
C   SEE LATER COMMENTS ABOUT FOLLOWING STATEMENTS
C   QRTL1=QRT
C   QRT=SUMRR*SUMRR+SUMRI*SUMRI
C   O1STAR CALCULATION
C   O1STAR=O1STAR+(TB(1)*TD(1)+TB(2)*TD(2)+TC(1)*TE(1)+TC(2)*TE(2))
C   1 *T(2)*4.0+4.0*T(4)*(TD(1)*TE(1)+TD(2)*TE(2))
C   IF (N.LT.3) GO TO 12
C   O2STAR CALCULATION
F1=TF(1)*TF(2)*TF(2)+TG(1)*TG(1)+TG(2)*TG(2)
F2=TB(1)*TF(1)+TB(2)*TF(2)+TC(1)*TG(1)+TC(2)*TG(2)
F3=TD(1)*TG(1)+TD(2)*TG(2)+TE(1)*TF(1)+TE(2)*TF(2)
ENL1=EN-1,0
COF1=2.50*((EN-2,0)*ENL1-3,0)*((EN-2,0)*ENL1-3,0)*(2.0*EN-3,0)/
((EN-2,0)*ENL1*(2,0*EN-1,0)*(2,0*EN-5,0))
COF2=7.50*((EN-2,0)*(EN+1,0)/(2.0*EN-1,0)
COF3=15.0/ENL1
O2STAR=O2STAR+COF1*F1+COF2*F2+COF3*F3
12 CONTINUE
QEXT=QEXT+T(3)*(TB(1)+TC(1))
T(4)=TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2)
QSCAT=QSCAT+T(3)*T(4)
T(2)=EN*EN+1,0
T(1)=T(3)/T(2)
DO 13 J=1,IT
PI3J=PI(3,J)
TAU3J=TAU(3,J)
ELTRMX(1,J)=ELTRMX(1,J)+T(1)*(TB(1)*PI3J+TC(1)*TAU3J)
ELTRMX(2,J)=ELTRMX(2,J)+T(1)*(TB(2)*PI3J+TC(2)*TAU3J)
ELTRMX(3,J)=ELTRMX(3,J)+T(1)*(TC(1)*PI3J+TB(1)*TAU3J)
ELTRMX(4,J)=ELTRMX(4,J)+T(1)*(TC(2)*PI3J+TB(2)*TAU3J)
13 CONTINUE
IF (N.LT.5) GO TO 14
QRTR=ABS((QRT-QRTL1)/QRT)
C   TEST FOR CONVERGENCE ON QEXT, QSCA, AND QRADAR
IF ((T(4).LT. TOL).AND.(QRTR.LT. TOL)) GO TO 16
14 N=N+1
DO 15 J=1,IT
PI(1,J)=PI(2,J)
PI(2,J)=PI(3,J)
TAU(1,J)=TAU(2,J)
TAU(2,J)=TAU(3,J)
15 CONTINUE
FNAPP=FNAP
FNBPP=FNPB
FNAP=FNA
FNPB=FNB
GO TO 2
16 CONTINUE
DO 18 J=1,IT
DO 17 I=1,4
T(I)=ELTRMX(I,J)
17 CONTINUE
ELTRMX(1,J)=T(3)*T(3)+T(4)*T(4)
ELTRMX(2,J)=T(1)*T(1)+T(2)*T(2)
ELTRMX(3,J)=T(1)*T(3)+T(2)*T(4)
ELTRMX(4,J)=(T(2)*T(3)-T(4)*T(1))/2.0
PFNZR0=(ELTRMX(1,J)+ELTRMX(2,J))/2.0
IF (IAPXCT.EQ.1) GO TO 20
P(J)=PFNZR0
18 CONTINUE
ELTRMX(2,J) IS THE VERTICAL COMPONENT SCATTERING I1 (EYE1)
ELTRMX(1,J) IS THE HORIZONTAL COMPONENT SCATTERING I2 (EYE2)
ELTRMX(3,J) IS EQUIVALENT TO EYE3
ELTRMX(4,J) IS EQUIVALENT TO -1.0*EYE4
T(1)=2.0*RX*RX
SGT=QEXT*T(1)
SGS=QSCAT*T(1)
O1STAR=3.0*O1STAR/(X*X*SGT)
O2STAR=4.0*O2STAR/(X*X*SGT)

```

MIEGX410
MIEGX420
MIEGX430
MIEGX440
MIEGX450
MIEGX460
MIEGX470
MIEGX480
MIEGX490
MIEGX500
MIEGX510
MIEGX520
MIEGX530
MIEGX540
MIEGX550
MIEGX560
MIEGX570
MIEGX580
MIEGX590
MIEGX600
MIEGX610
MIEGX620
MIEGX630
MIEGX640
MIEGX650
MIEGX660
MIEGX670
MIEGX680
MIEGX690
MIEGX700
MIEGX710
MIEGX720
MIEGX730
MIEGX740
MIEGX750
MIEGX760
MIEGX770
MIEGX780
MIEGX790
MIEGX800
MIEGX810
MIEGX820
MIEGX830
MIEGX840
MIEGX850
MIEGX860
MIEGX870
MIEGX880
MIEGX890
MIEGX900
MIEGX910
MIEGX920
MIEGX930
MIEGX940
MIEGX950
MIEGX960
MIEGX970
MIEGX980
MIEGX990
MIEGX000
MIEGX010
MIEGX020
MIEGX030
MIEGX040
MIEGX050
MIEGX060
MIEGX070
MIEGX080
MIEGX090
MIEGX100

```

SGR=(SUMRR+SUMRR+SUMRI*SUMRI)*RX*RX          MIEGX110
QTD=(SGT)                                     MIEGX120
QSD=(SGS)                                     MIEGX130
QRD=(SGR)                                     MIEGX140
Q1STRD=(Q1STAR)                               MIEGX150
Q2STRD=(Q2STAR)                               MIEGX160
C   LOOP FOR CALCULATION OF PFN AT ZERO DEGREES - FOR GPHASX
IAPXCT=1                                      MIEGX170
IT=1                                           MIEGX180
GO TO 21                                      MIEGX190
20 IT=ITT                                     MIEGX200
RETURN                                         MIEGX210
1000 FORMAT (52H V GT 20000 ERROR IN CONTINUED FRACTIONS MIE ROUTINE,
1 11H ** ALPHA = ,E12.6,6H EM = ,E12.6,7H CAY = ,E12.6/
2 1X,54H IT IS SUGGESTED THAT TOL=1.E-06 FOR SINGLE PRECISION. ) MIEGX230
END                                            MIEGX240
                                              MIEGX250
                                              MIEGX260

```

SUBROUTINE WATER(WAVE, EMT, CAYT, TMCHUR, RHODEN)
 COMMON /IO/, IGIN, IOUT, NUNIT, IEO, NEOU
 REAL LAMBDA(169), NSUBR(169), NSUBI(169)
 DIMENSION TEMP(7), DENS(7)
 THIS PROGRAM SEARCHES HALE AND QUERRY TABLE FOR REFRACTION VS.
 WAVELENGTH (APPLIED OPTICS, VOL. 12, NO. 3, MARCH 1973, PG 555)
 AND THE DENSITY VS. TEMPERATURE (HNDBK OF CHEM AND PHYS).
 IF THE INPUT VALUES (TMCHUR AND WAVE) ARE NOT IN THE TABLES,
 A LINEAR INTERPOLATION IS COMPUTED. VALUES ACCURATE TO THREE PLACES.
 INTEGER P,POINT,H
 TABLES FOLLOW FOR 60 LINES
 DATA LAMBDA/0.200,0.225,0.250,0.275,0.300,0.325,0.350,0.375,0.400
 1. 0.425,0.450,0.475,0.500,0.525,0.550,0.575,0.600,0.625,0.650
 2. 0.675,0.700,0.725,0.750,0.775,0.800,0.825,0.850,0.875,0.900
 3. 0.925,0.950,0.975,1.000,1.200,1.400,1.600,1.800,2.000,2.200
 4. 2.400,2.600,2.650,2.700,2.750,2.800,2.850,2.900,2.950,3.000
 5. 3.050,3.100,3.150,3.200,3.250,3.300,3.350,3.400,3.450,3.500
 6. 3.600,3.700,3.800,3.900,4.000,4.100,4.200,4.300,4.400,4.500
 7. 4.600,4.700,4.800,4.900,5.000,5.100,5.200,5.300,5.400,5.500
 8. 5.600,5.700,5.800,5.900,6.000,6.100,6.200,6.300,6.400,6.500
 9. 6.600,6.700,6.800,6.900,7.000,7.100,7.200,7.300,7.400,7.500
 X. 7.600,7.700,7.800,7.900,8.000,8.200,8.400,8.600,8.800,9.000
 1. 9.200,9.400,9.600,9.800,10.00,10.50,11.00,11.50,12.00,12.50
 2. 13.00,13.50,14.00,14.50,15.00,15.50,16.00,16.50,17.00,17.50
 3. 18.00,18.50,19.00,19.50,20.00,21.00,22.00,23.00,24.00,25.00
 4. 26.00,27.00,28.00,29.00,30.00,32.00,34.00,36.00,38.00,40.00
 5. 42.00,44.00,46.00,48.00,50.00,50.00,50.00,50.00,50.00,50.00
 6. 110.0,120.0,130.0,140.0,150.0,160.0,170.0,180.0,190.0,200.0
 /
 DATA NSUBR/1.396,1.373,1.362,1.354,1.349,1.346,1.343,1.341,1.339
 1. 1.338,1.337,1.336,1.335,1.334,1.333,1.332,1.331
 2. 1.331,1.331,1.330,1.330,1.330,1.329,1.329,1.328,1.328
 3. 1.328,1.327,1.327,1.327,1.324,1.321,1.317,1.312,1.306
 4. 1.279,1.242,1.219,1.188,1.157,1.142,1.149,1.201,1.292
 5. 1.426,1.467,1.483,1.478,1.467,1.450,1.432,1.420,1.410
 6. 1.385,1.374,1.364,1.357,1.351,1.346,1.342,1.338,1.334
 7. 1.330,1.330,1.330,1.328,1.325,1.322,1.317,1.312,1.305
 8. 1.289,1.277,1.262,1.248,1.265,1.319,1.363,1.357,1.347
 9. 1.334,1.329,1.324,1.321,1.317,1.314,1.312,1.309,1.307
 X. 1.302,1.299,1.297,1.294,1.291,1.286,1.281,1.275,1.269
 1. 1.255,1.247,1.239,1.229,1.218,1.185,1.153,1.126,1.111
 2. 1.146,1.177,1.210,1.241,1.270,1.297,1.325,1.351,1.376
 3. 1.423,1.443,1.461,1.476,1.480,1.487,1.500,1.511,1.521
 4. 1.539,1.545,1.549,1.551,1.551,1.546,1.536,1.527,1.522
 5. 1.522,1.530,1.541,1.555,1.587,1.703,1.821,1.886,1.924
 6. 1.966,2.004,2.036,2.056,2.069,2.081,2.094,2.107,2.119,2.130
 /
 DATA NSURI/1.10E-7,4.90E-8,3.35E-8,2.35E-8,1.60E-8,1.08E-8,6.50E-9
 1. 3.50E-9,1.86E-9,1.30E-9,1.02E-9,9.35E-10,1.00E-9,9.132E-9,1.96E-9
 2. 3.60E-9,1.09E-8,1.39E-8,1.64E-8,2.23E-8,3.35E-8,9.15E-8,1.56E-7
 3. 1.48E-7,1.25E-7,1.82E-7,2.93E-7,3.91E-7,4.86E-7,1.06E-6,2.93E-6
 4. 3.48E-6,2.89E-6,9.89E-6,1.38E-4,8.55E-5,1.15E-4,1.10E-3,2.89E-4
 5. 9.56E-4,3.17E-3,6.70E-3,1.90E-2,5.90E-2,1.15E-1,1.85E-1,2.68E-1
 6. 2.98E-1,2.72E-1,2.40E-1,1.92E-1,1.35E-1,9.24E-2,6.10E-2,3.68E-2
 7. 2.61E-2,1.95E-2,1.32E-2,9.40E-3,5.15E-3,3.60E-3,3.40E-3,3.80E-3
 8. 4.60E-3,5.62E-3,6.88E-3,8.45E-3,1.03E-2,1.34E-2,1.47E-2,1.57E-2
 9. 1.50E-2,1.37E-2,1.24E-2,1.11E-2,1.01E-2,9.80E-3,1.03E-2,1.16E-2
 X. 1.42E-2,2.03E-2,3.30E-2,6.22E-2,1.07E-1,1.31E-1,8.80E-2,5.70E-2
 1. 4.49E-2,3.92E-2,3.56E-2,3.37E-2,3.27E-2,2.32E-2,2.32E-2,3.20E-2,3.20E-2
 2. 3.21E-2,3.22E-2,3.24E-2,3.26E-2,3.28E-2,3.31E-2,3.35E-2,3.39E-2
 3. 3.43E-2,3.51E-2,3.61E-2,3.72E-2,3.85E-2,3.99E-2,4.15E-2,4.33E-2
 4. 4.54E-2,4.79E-2,5.08E-2,6.62E-2,9.68E-2,1.42E-1,1.99E-1,2.59E-1
 5. 3.05E-1,3.43E-1,3.70E-1,3.88E-1,4.02E-1,4.14E-1,4.22E-1,4.28E-1
 6. 4.29E-1,4.29E-1,4.26E-1,4.21E-1,4.14E-1,4.04E-1,3.93E-1,3.82E-1
 7. 4.73E-1,3.67E-1,3.61E-1,3.56E-1,3.50E-1,3.44E-1,3.38E-1,3.33E-1
 8. 3.28E-1,3.24E-1,3.29E-1,3.43E-1,3.61E-1,3.85E-1,4.09E-1,4.36E-1
 9. 4.62E-1,4.88E-1,5.14E-1,5.87E-1,5.76E-1,5.47E-1,5.36E-1,5.32E-1
 X. 5.31E-1,5.26E-1,5.14E-1,5.00E-1,4.95E-1,4.96E-1,4.97E-1,4.99E-1
 1. 5.01E-1,5.04E-1/
 ALTERNATE FORM OF ABOVE DATA STMT DUE TO EXCESS CONTINUATION CARDS WAT00700

```

DATA NSUBI(I), I=1,143) WAT00710
+ /1.10E-7, 4.90E-8, 3.35E-8, 2.35E-8, 1.60E-8, 1.08E-8, 6.50E-9WAT00720
1, 3.50E-9, 1.86E-9, 1.30E-9, 1.02E-9, 9.35E-10, 1.00E-9, 1.32E-9, 1.96E-9WAT00730
2, 3.60E-9, 1.09E-8, 1.39E-8, 1.64E-8, 2.23E-8, 3.35E-8, 9.15E-8, 1.56E-7WAT00740
3, 1.48E-7, 1.25E-7, 1.82E-7, 2.93E-7, 3.91E-7, 4.86E-7, 1.06E-6, 2.93E-6WAT00750
4, 3.48E-6, 2.89E-6, 9.89E-6, 1.38E-4, 6.55E-5, 1.15E-4, 1.10E-3, 2.89E-4WAT00760
5, 9.56E-4, 3.17E-3, 6.70E-3, 1.90E-2, 5.90E-2, 1.15E-1, 1.85E-1, 2.68E-1WAT00770
6, 2.98E-1, 2.72E-1, 2.40E-1, 1.92E-1, 1.35E-1, 3.24E-2, 6.10E-2, 3.68E-2WAT00780
7, 2.61E-2, 1.95E-2, 1.32E-2, 9.40E-3, 5.15E-3, 3.60E-3, 3.40E-3, 3.80E-3WAT00790
8, 4.60E-3, 5.62E-3, 6.88E-3, 8.45E-3, 1.03E-2, 1.34E-2, 1.47E-2, 1.57E-2WAT00800
9, 1.50E-2, 1.37E-2, 1.24E-2, 1.11E-2, 1.01E-2, 9.80E-3, 1.03E-2, 1.16E-2WAT00810
X, 1.42E-2, 2.03E-2, 3.30E-2, 6.22E-2, 1.07E-1, 1.31E-1, 8.80E-2, 5.70E-2WAT00820
1, 4.49E-2, 3.92E-2, 3.56E-2, 3.37E-2, 3.27E-2, 3.22E-2, 3.20E-2, 3.20E-2WAT00830
2, 3.21E-2, 3.22E-2, 3.24E-2, 3.26E-2, 3.28E-2, 3.31E-2, 3.35E-2, 3.39E-2WAT00840
3, 3.43E-2, 3.51E-2, 3.61E-2, 3.72E-2, 3.85E-2, 3.99E-2, 4.15E-2, 4.33E-2WAT00850
4, 4.54E-2, 4.79E-2, 5.08E-2, 6.62E-2, 9.68E-2, 1.42E-1, 1.99E-1, 2.59E-1WAT00860
5, 3.05E-1, 3.43E-1, 3.70E-1, 3.88E-1, 4.02E-1, 4.14E-1, 4.22E-1, 4.28E-1WAT00870
6, 4.29E-1, 4.29E-1, 4.26E-1, 4.21E-1, 4.14E-1, 4.04E-1, 3.93E-1, 3.82E-1WAT00880
7, 3.73E-1, 3.67E-1, 3.61E-1, 3.56E-1, 3.50E-1, 3.44E-1, 3.38E-1, 3.33E-1/WAT00890
DATA NSUBI(I), I=144,169) / WAT00900
8, 3.28E-1, 3.24E-1, 3.29E-1, 3.43E-1, 3.61E-1, 3.85E-1, 4.09E-1, 4.36E-1WAT00910
9, 4.62E-1, 4.88E-1, 5.14E-1, 5.87E-1, 5.76E-1, 5.47E-1, 5.36E-1, 5.32E-1WAT00920
X, 5.31E-1, 5.26E-1, 5.14E-1, 5.00E-1, 4.95E-1, 4.96E-1, 4.97E-1, 4.99E-1WAT00930
1, 5.01E-1, 5.04E-1, / WAT00940
DATA TEMP/273., 278., 283., 288., 293., 298., 303., /, DENS/ WAT00950
1, 0.999841, 0.999965, 0.999700, 0.999099, 0.998203, 0.997044, WAT00960
2, 0.995994/ WAT00970
EMT=0.0 WAT00980
CAYT=0.0 WAT00990
POINT=0 WAT01000
H=0 WAT01010
IF (WAVE.LT.0.2, OR, WAVE.GT.200.0, OR, TMCHUR.LT.273.0R, WAT01020
1, TMCHUR.GT.303.0) GO TO 11 WAT01030
C BINARY SEARCH WAT01040
L=1 WAT01050
H=125 WAT01060
1 POINT=((L+H)/2) WAT01070
TEST=ABS(LAMBDA(POINT)-WAVE) WAT01080
IF (TEST.LE.0.0001) GO TO 4 WAT01090
IF (WAVE.GT.LAMBDA(POINT)) GO TO 2 WAT01100
H=POINT WAT01110
GO TO 3 WAT01120
2 L=POINT+1 WAT01130
3 IF (L.NE.H) GO TO 1 WAT01140
L=L-1 WAT01150
C INTERPOLATION ROUTINE WAT01160
EMT=NSUBR(L)+(NSUBR(L+1)-NSUBR(L))*((WAVE-LAMBDA(L))/(LAMBDA(L+1)-WAT01170
1, -LAMBDA(L))) WAT01180
CAYT=NSUBI(L)+(NSUBI(L+1)-NSUBI(L))*((WAVE-LAMBDA(L))/(LAMBDA(L+1)-WAT01190
1, -LAMBDA(L))) WAT01200
GO TO 5 WAT01210
4 CONTINUE WAT01220
EMT=NSUBR(POINT) WAT01230
CAYT=NSUBI(POINT) WAT01240
C SEARCH TEMP VS DENS WAT01250
5 IF (TMCHUR.LT.273.0, OR, TMCHUR.GT.303.0) GO TO 11 WAT01260
L=1 WAT01270
H=7 WAT01280
6 P=((L+H)/2) WAT01290
TESTT=ABS(TEMP(P)-TMCHUR) WAT01300
IF (TESTT.LE.0.0001) GO TO 9 WAT01310
IF (TMCHUR.GT.TEMP(P)) GO TO 7 WAT01320
H=P WAT01330
GO TO 8 WAT01340
7 L=P+1 WAT01350
8 IF (L.NE.H) GO TO 6 WAT01360
L=L-1 WAT01370
RHODEN=DENS(L)+(DENS(L+1)-DENS(L))*(TMCHUR-TEMP(L))/(TEMP(L+1)- WAT01380
1, TEMP(L)) WAT01390
GO TO 10 WAT01400

```

```
9 RHODEN=DENS(P)          WAT 1410
10 CONTINUE                 WAT 01420
11 GO TO 13                 WAT 01430
11 WRITE (IOUT,12) TMCHUR,WAVE
12 FORMAT (10H TEMP, OF E12.6,14H OR WAVE, OF E12.6,14H BEYOND PANWAT01450
1GE, 24H OF DATA IN WATER ROUTINE/22H EXECUTION TERMINATED )
13 STOP                      WAT 01460
13 RETURN                   WAT 01470
END                         WAT 01480
                             WAT 01490
```

```

SUBROUTINE GAUS(IT)
COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)      GAUS0010
COMMON /IO/ IIN,IOUT,NUNIT,IEO,NEQU                                GAUS0020
COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                         GAUS0030
WRITE (IOUT,11)                                                       GAUS0040
C   INITIALIZE ARRAY PC() USED FOR RUNNING SUMMATION                 GAUS0050
DO 1 I=1,IT                                                       GAUS0060
OL(I)=0.                                                       GAUS0070
1 PC(I)=0.E+00                                                       GAUS0080
C   LOOPS 2 AND 3 CALCULATE EXPANSION COEFS. FOR FUNCTION PSUM()    GAUS0090
C   VIA GAUSS-LEGENDRE QUADRATURE. THE COEFS. GO INTO ARRAY OL()
DO 3 I=1,IT                                                       GAUS0100
COF=W(I)*PSUM(I)                                                 GAUS0110
PLM1=C(I)                                                       GAUS0120
PLM2=1.                                                       GAUS0130
DO 2 LL=1,IT                                                       GAUS0140
L=LL-1                                                       GAUS0150
PL=PLM2                                                       GAUS0160
IF (LL.EQ.2) PL=PLM1                                           GAUS0170
IF (LL.LE.2) GO TO 2                                           GAUS0180
PL=2.*C(I)*PLM1-PLM2-(C(I)*PLM1-PLM2)/FLOAT(L)               GAUS0190
PLM2=PLM1                                                       GAUS0200
PLM1=PL                                                       GAUS0210
2 OL(LL)=OL(LL)+COF*PL*(FLOAT(L)+.5)                           GAUS0220
CONTINUE                                                       GAUS0230
DO 7 I=1,IT                                                       GAUS0240
II=I-1                                                       GAUS0250
7 WRITE (IOUT,8) II,OL(I)                                         GAUS0260
8 FORMAT (1H ,20X,I6,20X,1PE14.7)                               GAUS0270
FORMAT (1H ,1X,25X,1HL,20X,16HL-TH COEFFICIENT)                GAUS0280
RETURN                                                       GAUS0290
END                                                       GAUS0300
GAUS0310
GAUS0320

```

```

SUBROUTINE DIMER(NGO)
COMMON /IO/IIN,IOUT,NUNIT,IEQ,NEOU
GO TO (1,2,3),NGO
1 WRITE(IOUT,101)
101 FORMAT(1H,'**** THE INPUT VALUE OF IT IS GREATER '
1 ' THAN THE ARRAY DIMENSIONS ',/, 'CHANGE THE DIMENSIONS OF THE ',
2 ' FOLLOWING ARRAYS IN SUBS AND COMMON ',/, 'COMMON BLOCK ',10X,
3 'ARRAY(S)',/, 'AGXM',10X, 'C,H,OLT',/, 'PT2',11X,
4 'PC,OL,RMS,P$UMT,P',/, 'ALSO CHANGE THE VALUE OF JIMCK(1) ',
5 'IN THE DATA STMT TO AGREE WITH THE NEW ',/, 'DIMENSION LIMITS',
6 '/,1H ,55HTHE SECOND INDEX ON ARRAYS PI, TAU, AND ELTRMX IN MIEGX
7 ,1H ,56HMUST ALSO BE CHANGED AND ARRAYS P AND C CHANGED AS WELL.',)
STOP
2 WRITE (IOUT,102)
102 FORMAT(1H,'**** TOO MANY PARTICLE RADII FOR DIMENSION LIMITS: ',
1 'IN SUBS AND COMMON CHANGE THE FOLLOWING ',
2 'ARRAYS ',/, 'COMMON BLOCK ',10X, 'ARRAY(S)',/, 'PT1',11X,'F,R,FF',
3 'ARRAYS F AND R MUST BE CONSISTENT WITH THE FOLLOWING: ARRAY ',
4 'SIZE = 1 + 2**JDIMCK(2)',/, 'ARRAY FF MUST BE DIMENSIONED TO ONE',
5 'MORE THAN ARRAYS F AND R',/, 'ALSO CHANGE THE VALUE OF JDIMCK(2) '
6 'IN THE DATA STATEMENT')
STOP
3 WRITE (IOUT,103)
103 FORMAT (1H , 'THE DIMENSIONS OF F AND '
1 'R DO NOT AGREE WITH THE FOLLOWING: ',/, 'SIZE=1+2**JDIMCK(2) ',
2 'WHERE JDIMCK(2) APPEARS IN THE DATA STATEMENT',/, 'ALSO DIMENSION '
3 'ARRAY FF TO BE ONE MORE THAN ARRAYS F AND R')
STOP
END

```

```

BLOCK DATA
IF YOU CHANGE THE DIMENSIONS MAKE SURE THAT YOU ALSO CHANGE THE      BLK00010
DATA STATEMENT CONTAINING JDIMCK();                                BLK00020
JDIMCK(1)=ORDER OF QUADRATURE (INPUT IT), I.E. MAX SIZE OF DIMENSION      BLK00030
JDIMCK(2): USED IN COMPUTATION OF SIZE OF ARRAYS F AND R,                BLK00040
SIZE=1+2**JDIMCK(2), ARRAY FF SHOULD BE ONE MORE THAN F AND R.          BLK00050
JDIMCK(3) WILL BE CALCULATED.                                         BLK00060
THE OTHER ARRAYS ARE REUSED, SO                                         BLK00070
CHECK THE SUBROUTINE CALLS AND COMMON BLOCKS TO SEE IF                  BLK00080
ARRAYS HAVE BEEN RENAMED WHEN U CHANGE DIMENSIONS.                      BLK00090
COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                           BLK00100
COMMON /PT1/ FC(513),RC(513),DR(8),RR(9),FF(514)                      BLK00110
+ NRADI,PI,IDSMP,NKG,NHALV,NI                                         BLK00120
COMMON /PT2/ PC(65),OLC(65),RMS(65),PSUM(65),PSUMT(65),P(65)        BLK00130
COMMON /IO/ IGIN,IOUT,HUNIT,IEO,NEOU                               BLK00140
DATA IGIN,IOUT,HUNIT,IEO,NEOU /5,6,3*0/                            BLK00150
DATA JDIMCK /65,9,0/                                              BLK00160
EOSAEL OPTION: 65 PREDETERMINED ANGLES                                BLK00170
DATA W
+   / 0.0,  0.5,  1.0,  2.0,  3.0,  4.0,  5.0,  6.0,  8.0, 10.0,  BLK00180
+   12.0, 14.0, 16.0, 18.0, 20.0, 24.0, 28.0, 32.0, 36.0, 40.0,  BLK00190
+   44.0, 48.0, 52.0, 56.0, 60.0, 64.0, 68.0, 72.0, 76.0, 80.0,  BLK00200
+   84.0, 88.0, 92.0, 96.0, 100.0, 104.0, 108.0, 112.0, 116.0, 120.0,  BLK00210
+   124.0, 128.0, 132.0, 136.0, 140.0, 142.0, 144.0, 146.0, 148.0, 150.0,  BLK00220
+   152.0, 154.0, 156.0, 158.0, 160.0, 162.0, 164.0, 166.0, 168.0, 170.0,  BLK00230
+   172.0, 174.0, 176.0, 178.0, 180.0/                                BLK00240
END

```

```

PROGRAM FLASH1
COMMON/IOUNT/I0IN,I0OUT
COMMON/CONST/PI
COMMON/IDATA/RDBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
*          RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLLOC,
*          TIME,ISRC,RCTSEE,RCTLOC
PI=3.14159
I0IN=5
I0OUT=6
C**** NOTE ICHK=-1 DEFAULTS USETR INPUT TO WAVE1
ICHK=0
WAVE1=.55
CALL FLASH(WAVE1,ICHK)
WRITE(I0OUT,100) WAVE1,ICHK
100 FORMAT(5X,13HEOSAEL OUTPUT/,
*5X,13HWAVELENGTH = ,F6.1,1X,11HMICROMETERS,/,
*5X,7HICHK = ,I4)
STOP
END

```

DRIV0010
DRIV0020
DRIV0030
DRIV0040
DRIV0050
DRIV0060
DRIV0070
DRIV0080
DRIV0090
DRIV0100
DRIV0110
DRIV0120
DRIV0130
DRIV0140
DRIV0150
DRIV0160
DRIV0170
DRIV0180

```

SUBROUTINE FLASH(WAVE1, ICHK)                                FLASH010
C*****PROGRAM FLASH                                         *FLASH020
C*                                                       *FLASH030
C*           EOSAEL80                                         *FLASH040
C*****DIMENSION FLASH1(3),FLASH2(3),IC1(3),IC2(3),IGUN(3)    *FLASH050
COMMON/IOUNT,I0IN,I0OUT                                         FLASH060
COMMON/CONST/PI                                              FLASH070
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,   FLASH080
*          RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,   FLASH090
*          TIME,ISRC,RCTSEE,RCTLOC                           FLASH100
* DATA FLASH1/3.85,2.55,0.0/                                  FLASH110
* DATA FLASH2/0.15,0.15,0.0/                                  FLASH120
* DATA IC1/2HH0,2H1-,2HUN/                                    FLASH130
* DATA IC2/2HW,,2H55,2HKN/                                    FLASH140
* DATA IGUN/05,100,0/                                       FLASH150
* DATA 1H1,20X,40(2H**),/,21X,1H*,34X,13HPROGRAM FLASH,31X,1H*,/,  FLASH160
98 FORMAT(//,21X,30H*****PROGRAM FLASH OUTPUT****,/,21X,40(2H--))  FLASH170
99 FORMAT(21X,20H*****END OF RUN****,/,21X,40(2H--))        FLASH180
100 FORMAT(1H1,20X,40(2H**),/,21X,1H*,34X,13HPROGRAM FLASH,31X,1H*,/,  FLASH190
*1H*,37X,BHEOSAEL80,33X,1H*,/,21X,40(2H**))        FLASH200
101 FORMAT(//,21X,15H*****INPUT****,/,21X,40(2H--))        FLASH210
102 FORMAT(21X,14HSCENARIO DATA:,/,21X,14HREFERENCE TIME,1X,F8.3,   FLASH220
*1X,3HSEC)                                                 FLASH230
103 FORMAT(21X,9HOSERVER:,18X,7HTARGET:,20X,7HSOURCE:)      FLASH240
104 FORMAT(24X,6HX(OBS),3X,F6.1,2H M,10X,6HX(TAR),3X,F6.1,2H M,10X,  FLASH250
*6HX(SRC),3X,F6.1,2H M,/,                                FLASH260
*24X,6HY(OBS),3X,F6.1,2H M,10X,6HY(TAR),3X,F6.1,2H M,10X,  FLASH270
*6HY(SRC),3X,F6.1,2H M,/,                                FLASH280
*24X,6HZ(OBS),3X,F6.1,2H M,10X,6HZ(TAR),3X,F6.1,2H M,10X,  FLASH290
*6HZ(SRC),3X,F6.1,2H M)                                 FLASH300
105 FORMAT(48X,11HORIENTATION,1X,F6.1,1X,3HDEG,8X,          FLASH310
*10HEVENT TIME,1X,F6.3,1X,3HSEC,/,67X,                  FLASH320
*10HCCW X-AXIS)                                            FLASH330
106 FORMAT(21X,25HDETECTOR CHARACTERISTICS:,/,             FLASH340
*24X,13HFIELD OF VIEW,12X,F6.1,1X,7HDEGREES,/,          FLASH350
*24X,10HWAVELENGTH,15X,F6.1,1X,11HMICROMETERS,/,       FLASH360
*24X,21HRESOLUTION CRITERIA--,/,                      FLASH370
*26X,17H(A) FOR DETECTION,6X,F6.3,1X,12HMILLIRADIANS,/,  FLASH380
*26X,15H(B) FOR LOCK ON,8X,F6.3,1X,12HMILLIRADIANS,/,  FLASH390
*24X,24HRECOVERY TIME (R=100M)--/,                   FLASH400
*26X,17H(A) FOR DETECTION,6X,F6.1,1X,7HSECONDS,/,     FLASH410
*26X,15H(B) FOR LOCK ON,8X,F6.1,1X,7HSECONDS)          FLASH420
107 FORMAT(21X,23HTARGET CHARACTERISTICS:,/               FLASH430
*24X,6HLENGTH,19X,F6.1,1X,6HMETERS,/,                 FLASH440
*24X,5HWIDTH,20X,F6.1,1X,6HMETERS,/,                 FLASH450
*24X,6HHEIGHT,19X,F6.1,1X,6HMETERS,/,                FLASH460
*24X,19HEXPOSURE CRITERIA--,/,                     FLASH470
*26X,17H(A) FOR DETECTION,6X,F6.1,1X,7HPERCENT,/,    FLASH480
*26X,15H(B) FOR LOCK ON,8X,F6.1,1X,7HPERCENT)        FLASH490
108 FORMAT(21X,23HSOURCE CHARACTERISTICS:,/,             FLASH500
*24X,4HTYPE,22X,I4,2HMM,1X,2A2,/,                  FLASH510
*24X,22HFLASH (VISIBLE) RADIUS,4X,F6.3,1X,6HMETERS,/,  FLASH520
*24X,24HFLASH (VISIBLE) DURATION,2X,F6.3,1X,7HSECONDS)  FLASH530
109 FORMAT(24X,27H*****PROGRAM FLASH END****,/,21X,40(2H--),1H1)  FLASH540
IWRIT=1                                                    FLASH550
IFLAG=0                                                    FLASH560
C*****READ IN DATA                                         FLASH570
1 WRITE(I0OUT,100)
CALL DATRD(IWRIT,IFLAG)
IF(ICHK.EQ.-1)WAVE=WAVE1
WAVE1=WAVE
IF(IFLAG.EQ.4)GO TO 9999
JSRC=ISRC
IF(ISRC.LT.1.OR.ISRC.GT.2)JSRC=3
FLASHR=FLASH1(JSRC)
FLASHT=FLASH2(JSRC)
WRITE(I0OUT,101)
WRITE(I0OUT,102)TIME
WRITE(I0OUT,103)
WRITE(I0OUT,104)(ROBS(I),RTAR(I),RSRC(I),I=1,3)  FLASH680
FLASH690
FLASH700

```

```
      WRITE( IOOUT, 105 ) TARDEG, TIMGUN          FLASH710
      WRITE( IOOUT, 106 ) FOV, WAVE, RADSEE, RADLOC, RCTSEE, RCTLOC   FLASH720
      WRITE( IOOUT, 107 ) TARLEN, TARWID, TARHGT, PCTSEE, PCTLOC    FLASH730
      WRITE( IOOUT, 108 ) IGUNK( JSRC ), IC1( JSRC ), IC2( JSRC ), FLASHR, FLASHT   FLASH740
      CALL GETIM( FLASHR, FLASHT, ISTOP, ISEE, ILOC, TIMLEF, TIMNOL )  FLASH750
      WRITE( IOOUT, 98 )                           FLASH760
      CALL DATWT( ISTOP, ISEE, ILOC, TIMLEF, TIMNOL )  FLASH770
      WRITE( IOOUT, 99 )                           FLASH780
      GO TO 1                                     FLASH790
 9999 WRITE( IOOUT, 109 )                      FLASH800
C****DEFINE EOSAEL OUTPUT
      ICHK=ISEE+1                                FLASH810
      STOP                                         FLASH820
      END                                           FLASH830
                                                FLASH840
```

```

SUBROUTINE DATRD(IWRIT,IFLAG)                                DTRD0010
C*****SUBROUTINE DATRD                                         /DTRD0020
C/*          SUBROUTINE DATRD                                     */DTRD0030
C/*          FLASH MODULE                                       */DTRD0040
C/*          EOSAEL80                                         */DTRD0050
C*****THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS   /DTRD0060
C THE SMOKE<EOSAEL> AND GRNAD<EOSAEL> MODULES                         DTRD0070
C*****INPUTS                                                 DTRD0080
C EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,           DTRD0100
C FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.             DTRD0110
C PER FIELD BEGINNING IN COL 11.  THE CARDS ARE NOT ORDER        DTRD0120
C DEPENDENT.                                                 DTRD0130
NAME          IGNORED                                         DTRD0140
SCEN          SCENARIO REFERENCE TIME(SEC)                      DTRD0150
              SOURCE TYPE CODE (1=105MM 2=100MM)                  DTRD0160
OBSC          TIME ISRC                                         DTRD0170
OBSC          ROBS(3)                                         DTRD0180
TARC          RTAR(3)                                         DTRD0190
TARC          TARDEG                                         DTRD0200
SRCC          RSRC(3)                                         DTRD0210
SRCC          TIMGUN                                         DTRD0220
DCHR          FOV                                            DTRD0230
DCHR          WAVE                                           DTRD0240
DCHR          RADSEE                                         DTRD0250
DCHR          RADLOC                                         DTRD0260
DCHR          RCTSEE                                         DTRD0270
DCHR          RCTLOC                                         DTRD0280
DCHR          RADLOC                                         DTRD0290
DCHR          RADSEE                                         DTRD0300
DCHR          RCTSEE                                         DTRD0310
DCHR          RCTLOC                                         DTRD0320
DCHR          RADSEE                                         DTRD0330
DCHR          RCTSEE                                         DTRD0340
DCHR          RCTLOC                                         DTRD0350
DCHR          RADSEE                                         DTRD0360
DCHR          RCTSEE                                         DTRD0370
DCHR          RCTLOC                                         DTRD0380
TCHR          TARLEN                                         DTRD0390
TCHR          TARWID                                         DTRD0400
TCHR          TARHGT                                         DTRD0410
PCTSEE         PCTSEE                                         DTRD0420
PCTSEE         FRACTION OF EXPOSURE NEEDED FOR DETECTION(DTRD0430
              (PERCENT)                                         DTRD0440
PCTLOC         PCTLOC                                         DTRD0450
PCTLOC         FRACTION OF EXPOSURE NEEDED FOR LOCK ON(DTRD0460
              (PERCENT)                                         DTRD0470
GO            SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT DTRD0480
DONE          END OF JOB.                                      DTRD0490
C*****COMMON/IOUNT,IOUTN,IOUTT                               DTRD0500
COMMON/IOUNT,IOUTN,IOUTT
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
*          RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
*          TIME,ISRC,RCTSEE,RCTLOC                           DTRD0510
*          DIMENSION IR1(18),IR1(2),R1(7),INAME(35)          DTRD0520
DATA IR/2HNA,2HME,2HSC,2HEN,2HOB,2HSC,2HTA,2HRC,2HSR,2HCC,
*          2HDC,2HHR,2HTC,2HHR,2HGO,2H ,2HDD,2HNE/          DTRD0530
*          100 FORMAT(21X,20H****CARD INPUT****,/,21X,40(2H--)) DTRD0540
101 FORMAT(2A2,6X,7F10.3)                                     DTRD0550
102 FORMAT(21X,2A2,6X,7F10.3)                                     DTRD0560
103 FORMAT(2A2,6X,35A2)                                       DTRD0570
104 FORMAT(21X,2A2,6X,35A2)                                     DTRD0580
C***** BEGINNING OF READ LOOP                                DTRD0590
C***** IF(IWRIT,EQ,0)GO TO 6                                 DTRD0600
C***** WRITE(IOUT,100)                                       DTRD0610
6 DO 10 I=1,9                                              DTRD0620
  IF(I,EQ,9)GO TO 90                                         DTRD0630
  IF(IFLAG,GT,0)GO TO 4                                     DTRD0640
  IFLAG=1                                                 DTRD0650
  READ(IOUTN,103)IR1(1),IR1(2),(INAME(J),J=1,35)          DTRD0660
  DTRD0670
  DTRD0680
  DTRD0690
  DTRD0700

```

```

IF(IWRIT.EQ.0)GO TO 4 DTRD0710
WRITE(IOOUT,104)IR1(1),IR1(2),(INAME(J),J=1,35) DTRD0720
4 READ(IOIN,101) IR1(1),IR1(2),(R1(J),J=1,7) DTRD0730
IF(IWRIT.EQ.0) GO TO 5 DTRD0740
WRITE(IOOUT,102) IR1(1),IR1(2),(R1(J),J=1,7) DTRD0750
5 IF(IR1(1).EQ.IR1(7).AND.IR1(2).EQ.IR1(8)) GO TO 998 DTRD0760
DTRD0770
DTRD0780
DTRD0790
C***** DTRD0800
C*** RELATING INPUT DATA TO VARIABLE NAMES. DTRD0810
C***** DTRD0820
IF(IR1(1).EQ.IR(1).AND.IR1(2).EQ.IR(2)) GO TO 10 DTRD0830
IF(IR1(1).EQ.IR(3).AND.IR1(2).EQ.IR(4)) GO TO 20 DTRD0840
IF(IR1(1).EQ.IR(5).AND.IR1(2).EQ.IR(6)) GO TO 30 DTRD0850
IF(IR1(1).EQ.IR(7).AND.IR1(2).EQ.IR(8)) GO TO 40 DTRD0860
IF(IR1(1).EQ.IR(9).AND.IR1(2).EQ.IR(10)) GO TO 50 DTRD0870
IF(IR1(1).EQ.IR(11).AND.IR1(2).EQ.IR(12)) GO TO 60 DTRD0880
IF(IR1(1).EQ.IR(13).AND.IR1(2).EQ.IR(14)) GO TO 70 DTRD0890
IF(IR1(1).EQ.IR(15).AND.IR1(2).EQ.IR(16)) GO TO 9999 DTRD0900
DTRD0910
DTRD0920
DTRD0930
DTRD0940
DTRD0950
DTRD0960
DTRD0970
DTRD0980
DTRD0990
DTRD1000
DTRD1010
DTRD1020
DTRD1030
DTRD1040
DTRD1050
DTRD1060
DTRD1070
DTRD1080
DTRD1090
DTRD1100
DTRD1110
DTRD1120
DTRD1130
DTRD1140
DTRD1150
DTRD1160
DTRD1170
DTRD1180
DTRD1190
DTRD1200
DTRD1210
DTRD1220
DTRD1230
DTRD1240
DTRD1250
DTRD1260
DTRD1270
DTRD1280
DTRD1290
DTRD1300
DTRD1310
DTRD1320
DTRD1330
DTRD1340
DTRD1350
DTRD1360
C***** DTRD1370
C*****CAUTION FOR TOO MANY CARDS DTRD1380
C***** DTRD1390
90 WRITE(IOOUT,106)
IFLAG=3
106 FORMAT(21X,17H*****CAUTION*****/, DTRD1400
*21X,56HMORÉ THAN 10 DATA CARDS ENTERED--REMAINING CARDS IGNORED> DTRD1410
GO TO 9999 DTRD1420
998 IFLAG=4 DTRD1430
9999 RETURN DTRD1440
END DTRD1450

```

```

SUBROUTINE DATWT(ISTOP, ISEE, ILOC, TIMLEF, TIMNOL) DAT00010
***** SUBROUTINE DATWT * DAT00020
C* FLASH MODULE * DAT00030
C* EOSAEL80 * DAT00040
***** DAT00050
C* DAT00060
C* DAT00070
COMMON/IOUNT/IOUT,IOOUT DAT00080
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE DAT00090
* RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLLOC, DAT00100
* TIME,ISRC,RCTSEE,RCTLOC DAT00110
* DIMENSION I1(3),I2(3) DAT00120
DATA I1/2H0,2HYE,2HPA/ DAT00130
DATA I2/2H ,2HS ,2HRT/
IANS1=ISEE+1 DAT00140
IF( ISEE.EQ.3)IANS1=3 DAT00150
IF( ILOC.EQ.0)IANS2=2 DAT00160
IF( ILOC.EQ.1)IANS2=1 DAT00170
IF( ILOC.EQ.2)IANS2=3 DAT00180
IF( ILOC.EQ.0)IANS2=2 DAT00190
IF( ILOC.EQ.3)IANS2=3 DAT00200
82 FORMAT(2IX,36HDETECTION EXPOSURE CRITERIA DEFEATED) DAT00210
83 FORMAT(2IX,38HDETECTION RESOLUTION CRITERIA DEFEATED) DAT00220
92 FORMAT(2IX,34HLOCK ON EXPOSURE CRITERIA DEFEATED) DAT00230
93 FORMAT(2IX,36HLOCK ON RESOLUTION CRITERIA DEFEATED) DAT00240
97 FORMAT(2IX,17HTARGET OBSCURED ?,1X,2A2,/, DAT00250
* 2IX,15HTARGET LOCKED ?,1X,2A2) DAT00260
98 FORMAT(2IX,10HTIME LEFT:,1X,F8.3,1X,7HSECONDS,/, DAT00270
* 2IX,13HTIME NO LOCK:,1X,F8.3,1X,7HSECONDS) DAT00280
99 FORMAT(2IX,25HPROGRAM FLASH--STOP CODE:,1X,3I1) DAT00290
100 FORMAT(2IX,11HSOURCE TYPE,1X,I2,1X,12HUNIDENTIFIED) DAT00300
101 FORMAT(2IX,17HINPUT WAVELENGTH:,1X,F6.1,1X,11HMICROMETERS, DAT00310
*25H IS OUT OF RANGE OF MODEL) DAT00320
102 FORMAT(2IX,37HGUNFLASH HAS NOT OCCURED YET: TIME = ,1X,F8.3,/,42X, DAT00330
*16HTIME OF FLASH = ,1X,F8.3) DAT00340
103 FORMAT(2IX,38HFLASH IS NOT IN DETECTOR FIELD OF VIEW) DAT00350
104 FORMAT(2IX,24HFLASH IS BEHIND TARGET--, DAT00360
*15H NO OBSCURATION) DAT00370
105 FORMAT(2IX,24HNORMAL PROGRAM EXECUTION) DAT00380
WRITE( IOUT,99)ISTOP,ISEE,ILOC DAT00390
IF( ISTOP.GT.1)GO TO 1 DAT00400
IF( ISTOP.GT.0)GO TO 10 DAT00410
WRITE( IOUT,100)ISRC DAT00420
GO TO 9999 DAT00430
10 WRITE( IOUT,101)WAVE DAT00440
GO TO 9999 DAT00450
1 WRITE( IOUT,97)I1(IANS1),I2(IANS1),I1(IANS2),I2(IANS2) DAT00460
IF( ISEE.LT.2.AND.ILOC.LT.2)GO TO 11 DAT00470
IF( ISEE.EQ.2)WRITE( IOUT,82) DAT00480
IF( ISEE.EQ.3)WRITE( IOUT,83) DAT00490
IF( ILOC.EQ.2)WRITE( IOUT,92) DAT00500
IF( ILOC.EQ.3)WRITE( IOUT,93) DAT00510
GO TO 9998 DAT00520
11 IGO=ISTOP-1 DAT00530
GO TO (2,3,4,5,6,7,8,9),IGO DAT00540
2 WRITE( IOUT,102)TIME,TIMGUN DAT00550
GO TO 9999 DAT00560
3 WRITE( IOUT,103) DAT00570
GO TO 9999 DAT00580
4 WRITE( IOUT,103) DAT00590
GO TO 9999 DAT00600
5 WRITE( IOUT,104) DAT00610
GO TO 9999 DAT00620
6 WRITE( IOUT,103) DAT00630
GO TO 9999 DAT00640
7 WRITE( IOUT,103) DAT00650
GO TO 9999 DAT00660
8 WRITE( IOUT,103) DAT00670
GO TO 9999 DAT00680
9 WRITE( IOUT,105) DAT00690
9998 WRITE( IOUT,98)TIMLEF,TIMNOL DAT00700

```

9999 RETURN
END

DAT00710
DAT00720

```

SUBROUTINE GETIM(FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)      GTM00010
***** SUBROUTINE GETIM                                         GTM00020
**          FLASH MODULE                                         *GTM00020
**          E0SAEL80                                         *GTM00040
***** REAL LOCSTO,LOCFRA                                         *GTM00050
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,        GTM00060
*          RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLLOC,       GTM00070
*          TIME,ISRC,RCTSEE,RCTLLOC                               GTM00080
COMMON/CONST/PI                                         GTM00090
***** ALL DISTANCES IN METERS ANGLES CONVERTED TO RADIANS(CCW A-AXES   GTM00100
DIMENSION DOTVEC(3),RHTVEC(3),RHFVEC(3),RFLASH(3)                   GTM00110
***** SCALAR(A,B)=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)                  GTM00120
SCALAR(A1,A2,A3,B1,B2,B3)=A1*B1+A2*B2+A3*B3                      GTM00130
***** DEFAULT AND CONVERT INPUT                                     GTM00140
ANGGUN=0.0                                         GTM00150
ELEGUN=0.0                                         GTM00160
GUNHT=0.0                                         GTM00170
GUNLEN=0.0                                         GTM00180
TARHHT=TARHGT/2.0                                         GTM00190
ANGTAR=TARDEC*(PI/180.0)                                         GTM00200
HAFFOV=(FOV/2.0)*(PI/180.0)                                         GTM00210
SEESTO=RADSEE/1000.0                                         GTM00220
LOCSTO=RADLOC/1000.0                                         GTM00230
SEEFRA=PCTSEE/100.0                                         GTM00240
LOCFRA=PCTLLOC/100.0                                         GTM00250
RADFL=FLASHR                                         GTM00260
DURTIM=FLASHT                                         GTM00270
DURSEE=10.0*RCTSEE                                         GTM00280
DURLOC=10.0*RCTLLOC                                         GTM00290
REACFA=SEEFRA-LOCFRA                                         GTM00300
***** INITIALIZE FLAGS *****
AGINSE=0.0                                         GTM00310
AGINLO=0.0                                         GTM00320
TIMLEF=0.0                                         GTM00330
TIMNOL=0.0                                         GTM00340
TIMGON=TIME-TIMGUN                                         GTM00350
ISEE=0                                         GTM00360
ILOC=0                                         GTM00370
ISTOP=0                                         GTM00380
IF(ISRC.LT.1.OR.ISRC.GT.2)GO TO 9999                 GTM00390
ISTOP=1                                         GTM00400
IWAVE=0                                         GTM00410
IF(WAVE.GE.0.40.AND.WAVE.LE.0.70)IWAVE=1             GTM00420
IF(WAVE.GE.8.00.AND.WAVE.LE.12.0)IWAVE=2            GTM00430
IF(IWAVE.EQ.0)GO TO 9999                           GTM00440
ISTOP=2                                         GTM00450
IF(TIMGON.LT.0.)GO TO 9999                           GTM00460
TIMLEF=DURTIM-TIMGON                                         GTM00470
***** TIMLEF IS DURATION LEFT OF FLASH                GTM00480
TIMNOL =TIMLEF+REACFA*DURTIM                           GTM00490
***** TIMNOL IS DURATION OF LOCK                         GTM00500
FLSHIF=RADFL                                         GTM00510
***** CALCULATE FLASH COORDINATES FROM GIVEN COORDINATES   GTM00520
RFLASH(1)=RSRC(1)+FLSHIF*COS(ANGGUN)*COS(ELEGUN)    GTM00530
RFLASH(2)=RSRC(2)+FLSHIF*SIN(ANGGUN)*COS(ELEGUN)    GTM00540
RFLASH(3)=RSRC(3)+FLSHIF*SIN(ELEGUN) + GUNHT         GTM00550
***** DEFINE DIRECTION OF TARGET UNIT VECTOR           GTM00560
DOTVEC(1)=COS(ANGTAR)                                 GTM00570
DOTVEC(2)=SIN(ANGTAR)                                 GTM00580
DOTVEC(3)=0.0                                         GTM00590
***** DEFINE OBSERVER-TARGET, OBSERVER-FLASH VECTORS   GTM00600
DO 10 I=1,3                                         GTM00610
RHTVEC(I)=RTAR(I)-ROBS(I)                            GTM00620
RHFVEC(I)=RFLASH(I)-ROBS(I)                           GTM00630
10 CONTINUE                                         GTM00640
***** ADD HALF TARGET HEIGHT                          GTM00650
RHTVEC(3)=RHTVEC(3)+TARHHT                           GTM00660
***** FIND LENGTHS                                     GTM00670
GTM00680
GTM00690
GTM00700

```

```

A1=RHTVECC(1) GTM00710
A2=RHTVECC(2) GTM00720
A3=RHTVECC(3) GTM00730
B1=RHFVECC(1) GTM00740
B2=RHFVECC(2) GTM00750
B3=RHFVECC(3) GTM00760
RHT=SQRT(SCALAR(A1,A2,A3,A1,A2,A3)) GTM00770
RHF=SQRT(SCALAR(B1,B2,B3,B1,B2,B3)) GTM00780
*****FIND COSINES BETWEEN VECTORS GTM00790
C1=DOTVEC(1) GTM00800
C2=DOTVEC(2) GTM00810
C3=DOTVEC(3) GTM00820
GAMMAD= SCALAR(C1,C2,C3,B1,B2,B3)/RHF GTM00830
GAMMAT= SCALAR(A1,A2,A3,B1,B2,B3)/(RHT*RHF) GTM00840
W= SCALAR(C1,C2,C3,A1,A2,A3)/RHT GTM00850
*****FIND TRIG FCNS OF THETA=HALF-ANGLE OF FLASH CONE GTM00860
SINTH=RADFL/RHF GTM00870
COSTH=SQRT(1.0-SINTH*SINTH) GTM00880
COSSQ=COSTH*COSTH GTM00890
COSINE=ABS(W) GTM00900
SINE=SQRT(1.0-COSINE**2) GTM00910
*****CALCULATE CYLINDRICAL TARGET LONGEST DIMENSION AS SEEN IN GTM00920
*****PLANE PERPENDICULAR TO RHFVEC GTM00930
TARDIM=TARLEN*SINE+TARWID*COSINE GTM00940
HAFDIM=TARDIM/2 GTM00950
THETA=ATAN2(SQRT(1.0-COSTH**2),COSTH) GTM00960
THETAT=ATAN2(SQRT(1.0-GAMMAT**2),GAMMAT) GTM00970
DELTH=THETAT-THETA GTM00980
ISTOP=3 GTM00990
IF(DELTH.GT.HAFFOV)GO TO 9999 GTM01000
*****IF OUT OF FIELD OF VIEW, RETURN. GTM01010
DISPLC=RHT*SIN(DELTH) GTM01020
PROJRH=RHT*GAMMAT GTM01030
ISTOP=4 GTM01040
IF(DISPLC.GT.HAFDIM)GO TO 99 GTM01050
ISTOP=5 GTM01060
IF(PROJRH.LT.RHF.AND.IWAVE.LT.2)GO TO 99 GTM01070
*****NO OBSCURATION IF TARGET IN FRONT OF FLASH SO RETURN GTM01080
*****FROM HERE ON GAMMAT NECESSARILY POSITIVE GTM01090
*****FIND INTERSECTIONS OF FLASH CONE WITH DOTVEC EXTENDED FROM TARGET GTM01100
*****GET COEFFICIENT OF QUADRATIC EQUATION FOR DISTANCE ALONG DOTVEC GTM01110
*****FROM TARGET TO FLASH CONE/RHT GTM01120
A=COSSQ-W-GAMMAD**2 GTM01130
BD2=COSSQ*W-GAMMAT*GAMMAT GTM01140
C=COSSQ-W-GAMMAT**2 GTM01150
B=BD2*2 GTM01160
*****IF DISCRIMINANT NEGATIVE, NO INTERSECTION GTM01170
DISCRM=BD2*BD2-A*C GTM01180
ISTOP=6 GTM01190
IF(DISCRM.LT.0.)GO TO 99 GTM01200
*****IF A=0, QUADRATIC FORMULA BLOWS UP. ACTUALLY HAVE LINEAR EQN. GTM01210
IF(ABS(A).GT.1.E-30)GOTO20 GTM01220
IF(ABS(B).LT.1.E-30)B=SIGN(1.E-30,B) GTM01230
SPL=-C/B*RHT GTM01240
ZPL=SPL*GAMMAD+PROJRH GTM01250
ISTOP=7 GTM01260
IF(ZPL.LE.0.) GO TO 99 GTM01270
*****REJECT IF SOLE INTERSEC IS W NEG CONE SHEET. MEANINGLESS GTM01280
DPL=ABS(SPL)*SINE GTM01290
IF(GAMMAT-COSTH)14,14,12 GTM01300
*****IF TAR OUTSIDE FLASH CONE GO TO 14, OTHERWISE 12 GTM01310
12 OBSCUR=HAFDIM+AMIN1(DPL,HAFDIM) GTM01320
GO TO 30 GTM01330
14 OBSCUR=AMAX1(0.,HAFDIM-DPL) GTM01340
GO TO 30 GTM01350
20 ROOT=SQRT(DISCRM) GTM01360
ROOT=SIGN(ROOT,A) GTM01370
*****ABOVE NOT REALLY NEEDED BUT NICER TO HAVE SPL.GT.SMI-SEE BELOW GTM01380
SPL=(-BD2+ROOT)/A *RHT GTM01390
SMI=(-BD2-ROOT)/A *RHT GTM01400

```

```

*****SPL,SMI ARE 2 LENGTHS ALONG DOTVEC OF INTERSEC PTS W FLASH CONE      GTM01410
*****ZPL,ZMI ARE Z COORDINATES OF INTERSECTIONS(ON FLASH CONE AXIS)      GTM01420
ZPL=SPL*GAMMAD+PROJRH                                              GTM01430
ZMI=SMI*GAMMAD+PROJRH                                              GTM01440
ISTOP=8                                                               GTM01450
IF(ZPL.LE.0.,AND,ZMI.LE.0.)GO TO 99                                     GTM01460
*****REJECT IF BOTH INTERCEPTS IN NEGATIVE CONE                         GTM01470
*****DPL,DMI ARE PROJECS OF SPL,SMI PERPEN TO RHTVEC                   GTM01480
*****THEIR MAGNITUDES ARE LIMITED TO HAFDIM                           GTM01490
DPL=SPL*SINE                                              GTM01500
DPL=AMIN1(DPL,HAFDIM)                                         GTM01510
DPL=AMAX1(DPL,-HAFDIM)                                         GTM01520
DMI=SMI*SINE                                              GTM01530
DMI=AMIN1(DMI,HAFDIM)                                         GTM01540
DMI=AMAX1(DMI,-HAFDIM)                                         GTM01550
OBSCUR=ABS(DPL-DMI)                                         GTM01560
IF(ZPL*ZMI.GT.0.)GOTO30                                         GTM01570
*****SKIP AROUND UNLESS BOTH SHEETS OF CONE INVOLVED                 GTM01580
IF(ZPL.LE.0.)DPL=SIGN(HAFDIM,DPL)                                     GTM01590
IF(ZMI.LE.0.)DMI=SIGN(HAFDIM,DMI)                                     GTM01600
OBSCUR=TARDIM-ABS(DPL-DMI)                                         GTM01610
*****ABOVE BRANCH RARE. INTERSEC W BOTH CONE SHEETS, NEG SHEET IGNORE. GTM01620
30 CONTINUE
SEEN=TARDIM-OBSCUR                                         GTM01630
ISTOP=9                                                       GTM01640
98 IF(IWAVE.EQ.2)GO TO 1                                         GTM01650
CALL VSBLC(ISEE,ILOC,SEEN,TARDIM,RHT,TIMLEF,SEESTO,LOCSTO,          GTM01660
*           SEEFR,LOCFR)                                         GTM01670
*           GO TO 99                                           GTM01680
1 CALL IRBLCK(RHF,ISEE,ILOC,AGINSE,AGINLO,DURSEE,DURLOC,TIMGON)    GTM01690
99 CONTINUE
TIMLEF =AMAX1(TIMLEF,AGINSE)                                         GTM01710
TIMNOL=AMAX1(TIMNOL,AGINLO)                                         GTM01720
9999 RETURN
END

```

```

SUBROUTINE IRBLCK(RHF, ISEE, ILOC, AGINSE, AGINLO, DURSEE, DURLOC, TIMGON)IRBLC010
*****IRBLC020
C*          SUBROUTINE IRBLC          *IRBLC030
C*          FLASH MODULE           *IRBLC040
C*          EOSAEL80                *IRBLC050
C*                                      *****IRBLC060
C*          INTEGER LOCIRM          IRBLC070
C*          SEEIRM=4.0               IRBLC080
C*          LOCIRM=5.0               IRBLC090
C*          SQRTRH=SQRT(RHF)         IRBLC100
C*          AGINSE=DURSEE/SQRTRH    IRBLC110
C*          AGINLO=DURLOC/SQRTRH    IRBLC120
C*****ABOVE WHOLLY EMPIRICAL FROM FIT TO TV TAPES AT 2 RANGES
C*          IF(AGINSE.GT.SEEIRM) AGINSE=SEEIRM   IRBLC130
C*          IF(AGINLO.GT.LOCIRM) AGINLO=LOCIRM   IRBLC140
C*          IF(AGINSE.GT.TIMGON) ISEE=1           IRBLC150
C*          IF(AGINLO.GT.TIMGON) ILOC=1           IRBLC160
C*          RETURN                         IRBLC170
C*          END                           IRBLC180
C*                                      IRBLC190

```

```

SUBROUTINE VSBLC(ISEE,ILOC,SEEN,TARDIM,RHT,TIMLEF,SEESTO,LOCSTO, VSBLC010
*          SEEFR,LOCFR) VSBLC020
***** VSBLC030
C*          SUBROUTINE VSBLC * VSBLC040
C*          FLASH MODULE * VSBLC050
C*          EOSAEL80 * VSBLC060
***** VSBLC070
REAL LOCRI,LOCFR,LOCSTO VSBLC080
C****SEEFR IS FRACTION THAT MUST BE SEEN,LIKEWISE FOR LOCK VSBLC090
C****SEESTO IS MIN ANGLE THAT MUST BE SEEN,LIKEWISE FOR LOCK VSBLC100
SEEANG=SEEN/RHT VSBLC110
TARANG=TARDIM/RHT VSBLC120
*****DEFINE SEEN ANGLE AND TARGET ANGLE VSBLC130
SEEcri=AMIN1(TARANG,SEESTO) VSBLC140
LOCcri=AMIN1(TARANG,LOCSTO) VSBLC150
IF(TIMLEF.LE.0.)GO TO 9999 VSBLC160
IF(SEEANG.LT.SEEFR*TARANG)GO TO 48 VSBLC170
ISEE=2 VSBLC180
GO TO 50 VSBLC190
48 IF(SEEANG.LT.SEEcri)GO TO 49 VSBLC200
ISEE=1 VSBLC210
GO TO 50 VSBLC220
49 CONTINUE VSBLC230
ISEE=3 VSBLC240
50 CONTINUE VSBLC250
IF(SEEANG.LT.LOCFR*TARANG)GO TO 98 VSBLC260
ILOC=2 VSBLC270
GO TO 9999 VSBLC280
98 IF(SEEANG.LT.LOCcri)GO TO 99 VSBLC290
ILOC=1 VSBLC300
GO TO 9999 VSBLC310
99 CONTINUE VSBLC320
ILOC=3 VSBLC330
9999 RETURN VSBLC340
END VSBLC350

```

EORUN , 4.0
 WAVL , 10.6 10.6
 VIS , 7.0
 TURB
 XSCALE , 3.0
 SMOKE
 DRTRAN
 LZTRAN
 CLTRAN
 SCREEN
 DVRLST , 2.0
 CLIMAT
 GRNADE
 GO
 PARM 0.1016 1.3E-04 0.0 400.0 5.0 512.0
 CN1 1.0 6.0E-13 1.0E-14 1.3E-14 2.7E-14 5.0E-14
 V1 1.0 0.93 0.93 0.93 0.93 0.93
 DVRY 0.400 500.0
 CN2 1.0 6.0E-13 1.0E-14 1.3E-14 2.7E-14 5.0E-14
 V2 1.0 0.93 0.93 0.93 0.93 0.93
 GU
 END
 FOG 1.
 HORIZ 0.4
 GO
 FOG 2.
 SLNH 0.133, 56.3
 GO
 FOG 3.
 HORIZ 0.4
 GO
 END
 MUNC 0.0 -50.0 0.
 OBSC 200. 0. 200.
 TARC -200. 0. 0.
 BART 5. 180. 0.
 OUTP 0. 0. 0.
 BURN 10. 0. 0.
 GU
 BURN 4. 0. 0.
 BART 5. 250. 5. 90.
 GO
 DONE
 END
 MET1 4. 2. 286.0 2. 2.0 0.0
 MET2 1. 53. 0. 0.15
 SOIL 2. 0. 0. 0.0
 CHAR 3. 6.8 0. 0.
 EXPL 1. 1. 1. 0.0
 OBSC 200.0 0.0 2.0
 TRNC -200.0 0.0 2.0
 RECO 200.0 0.0 2.0
 TIMS 1.0 71.0 2.0
 GO
 DONE
 END 8.55 15. 0.4
 END
 SEEK, 0.2, 0.0, 0.6
 TARG, -0.2, 0.0, 0.002
 CLST, 0.20, 0.40
 GO
 SEEK, 0.133, 0.0, 0.5
 GO
 SEEK, 0.0, 0.0, 0.3
 GO
 END
 TARY 1.1 1.0 2.0 0.24 2.3 81. 2. .000
 SENS .99 8. 1.

GO
 SENS .90 8. 1. 0. 0. 1.
 GO SENS .75 8. 1. 0. 0. 1.
 GO SENS .50 8. 1. 0. 0. 1.
 GO SENS .10 8. 1. 0. 0. 1.
 GO
 DONE
 SCRN 15. 400. 0.4 0. 90. 74. 2.
 DONE
 END
 UP0S, -0.0667, 0.0, 0.2
 CLDS, -0.2, 200.0, 40.0, 0.7, 1.0
 SP0S, -0.2, 0.0, 0.002
 BKGR, 50.0
 GRND, 50.0
 TEMP, 9.8
 GO
 UP0S, -0.1333, 0.0, 0.1
 GO
 END
 NAME
 OUTP
 OBSC 200.0 0.0 2.0
 MUNC -200.0 0.0 2.0 95.0 100.0 10.0
 TARO -200.0 +40. 50.0
 BART 5.0 400.0 50.0 90.0
 MUNT 1.0 0.793 14.3 1.0 0.0 4.7 0.07
 METR 50.0 2.0 220.0 4.0 20.0 0.0
 EXTC
 BURN
 MISC
 GO
 DONE
 END
 WAVL 1.06 1.06
 VIS ; 5.0 5.0 4.0
 BASCAT
 FCLOUD
 GO
 PART, 1. 5000. 1.
 SORC, -0.2, 0., -0.098, 90., 0., 50.
 DETR, 0.2, 0., -0.098, 90., 180., 1., 1.
 CLDS, -1., 0.2, .1, 0., 0., 0., 0.
 GRND, -0.1, 0.5
 PULS, .33, 0.
 GO
 END
 CP0S, 0.0, 0.0, 0.1
 RF0S, 0.2, 0.0, 0.002
 SP0S, -0.2, 0.0, 0.002
 AXES, 0.1, 0.2, 0.1
 CLDS, 5.0, 0.95, 4.0, 1.0, 9.8
 ATMO, 2.0, 9.8
 BKGR, 0.5, 50.0
 SANC, 80.0, 0.0
 LUND, 0.0
 GO
 END
 FREQ 35.0 35.0
 NMMW ; 2.0
 GO
 PATH, 0.4
 ATMO, 15.0 1013.25 6.44
 FOGD, 0.5
 RAIN, 5.0
 GO
 ATMO, -1.0 1015.2 7.7

```

RAIN,      0.0
SNOW,     5.0
GO
END
WVNUM    2010, 2710, 2.
LT4M
RESF
SPGT
GU
ENVR      3.          2.          2.          4.          1.          1.
EMIS      .100+01   .283+03   .950+00   .295+03
ATM       .650+02   .0.0       .0.0       .0.0
TARG      .400+00   .450+02   .900+02   .450+02
REFL      0.5        0.5        0.0        .500-01   .0
SENS      .200-02   .900+02   .270+03   .100+01   0.
GO
12
3.5,      0.78
3.6,      0.83
3.7,      0.87
3.8,      0.92
3.9,      0.96
4.0,      0.98
4.1,      0.97
4.3,      0.96
4.5,      0.95
4.7,      0.94
4.9,      0.93
5.0,      0.93
END
4  2 1 0 1 0 0 0 0 0.000 0.000 0.000 0.000 1.000
4
4  2 1 0 1 0 0 0 1 0.000 0.000
0
END

```

```
*****  
* ELECTRO-OPTICAL SYSTEMS *  
* ATMOSPHERIC EFFECTS *  
* LIBRARY *  
*****
```

INDIVIDUAL MODULES SELECTED	BEGINNING	ENDING
TURB XSCALE SMOKE DRTRAN LZTRAN SCREEN OVRCST GRNADE CLIMATE	943.396	943.396
WAVENUMBER(CH*-1)		
WAVELENGTH(MICRONS)	10.600	10.600
FREQUENCY(GHZ)	28301.886	28301.886

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

CLASS 1 FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS 2 FOG, HAZE AND MIST WITH VIS LT 3 KM.
CLASS 3 FOG, HAZE AND MIST WITH VIS LT 7 KM.
CLASS 4 FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS 5 DUST WITH VIS GE 3 KM.
CLASS 6 DUST WITH VIS GE 1 KM.
CLASS 7 DUST AND RAIN WITH VIS LT 1 KM.
CLASS 8 DUST AND RAIN WITH VIS LT 3 KM.
CLASS 9 DUST AND RAIN AND THUNDERSTORMS WITH VIS LT 1 KM.
CLASS 10 DUST AND RAIN AND THUNDERSTORMS WITH VIS LT 3 KM.
CLASS 11 SNOW WITH VIS LT 1 KM.
CLASS 12 SNOW WITH VIS LT 3 KM.
CLASS 13 SNOW AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 14 SNOW AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 15 NO WEATHER AND CEILING HEIGHT LT 300 M.
CLASS 16 NO WEATHER AND CEILING HEIGHT LT 1000 M.
CLASS 17 VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 VIS LT 3 KM AND CEILING HEIGHT LT 300 M.
CLASS 19 VIS LT 1000 M.
CLASS 20 CEILING HEIGHT LT 1000 M.
CLASS 21 CEILING HEIGHT LT 300 M.
CLASS 22 CEILING HEIGHT LT 1000 M.

EDSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS

EUROPEAN LOWLANDS									
CLASS NO.	FREQUENCY CLASS (%)	MEAN TEMP (°C)	MEAN RH (%)	MEAN VIS (km)	MEAN PRESS (mb)	MEAN CLOUD (km)	MEAN SIGHTDAY (%)	MEAN FREQUENCY (%)	DURING SPRING AT 03-09 (LST).
1	1-10	-1.0	85	10	1010	1.0	2.0	1.0	1-10
2	11-20	-0.5	85	10	1010	1.0	2.0	1.0	11-20
3	21-30	0.0	85	10	1010	1.0	2.0	1.0	21-30
4	31-40	0.5	85	10	1010	1.0	2.0	1.0	31-40
5	41-50	1.0	85	10	1010	1.0	2.0	1.0	41-50
6	51-60	1.5	85	10	1010	1.0	2.0	1.0	51-60
7	61-70	2.0	85	10	1010	1.0	2.0	1.0	61-70
8	71-80	2.5	85	10	1010	1.0	2.0	1.0	71-80
9	81-90	3.0	85	10	1010	1.0	2.0	1.0	81-90
10	91-100	3.5	85	10	1010	1.0	2.0	1.0	91-100
11	101-110	4.0	85	10	1010	1.0	2.0	1.0	101-110
12	111-120	4.5	85	10	1010	1.0	2.0	1.0	111-120
13	121-130	5.0	85	10	1010	1.0	2.0	1.0	121-130
14	131-140	5.5	85	10	1010	1.0	2.0	1.0	131-140
15	141-150	6.0	85	10	1010	1.0	2.0	1.0	141-150
16	151-160	6.5	85	10	1010	1.0	2.0	1.0	151-160
17	161-170	7.0	85	10	1010	1.0	2.0	1.0	161-170
18	171-180	7.5	85	10	1010	1.0	2.0	1.0	171-180
19	181-190	8.0	85	10	1010	1.0	2.0	1.0	181-190
20	191-200	8.5	85	10	1010	1.0	2.0	1.0	191-200
21	201-210	9.0	85	10	1010	1.0	2.0	1.0	201-210
22	211-220	9.5	85	10	1010	1.0	2.0	1.0	211-220
23	221-230	10.0	85	10	1010	1.0	2.0	1.0	221-230
24	231-240	10.5	85	10	1010	1.0	2.0	1.0	231-240
25	241-250	11.0	85	10	1010	1.0	2.0	1.0	241-250
26	251-260	11.5	85	10	1010	1.0	2.0	1.0	251-260
27	261-270	12.0	85	10	1010	1.0	2.0	1.0	261-270
28	271-280	12.5	85	10	1010	1.0	2.0	1.0	271-280
29	281-290	13.0	85	10	1010	1.0	2.0	1.0	281-290
30	291-300	13.5	85	10	1010	1.0	2.0	1.0	291-300
31	301-310	14.0	85	10	1010	1.0	2.0	1.0	301-310
32	311-320	14.5	85	10	1010	1.0	2.0	1.0	311-320
33	321-330	15.0	85	10	1010	1.0	2.0	1.0	321-330
34	331-340	15.5	85	10	1010	1.0	2.0	1.0	331-340
35	341-350	16.0	85	10	1010	1.0	2.0	1.0	341-350
36	351-360	16.5	85	10	1010	1.0	2.0	1.0	351-360
37	361-370	17.0	85	10	1010	1.0	2.0	1.0	361-370
38	371-380	17.5	85	10	1010	1.0	2.0	1.0	371-380
39	381-390	18.0	85	10	1010	1.0	2.0	1.0	381-390
40	391-400	18.5	85	10	1010	1.0	2.0	1.0	391-400
41	401-410	19.0	85	10	1010	1.0	2.0	1.0	401-410
42	411-420	19.5	85	10	1010	1.0	2.0	1.0	411-420
43	421-430	20.0	85	10	1010	1.0	2.0	1.0	421-430
44	431-440	20.5	85	10	1010	1.0	2.0	1.0	431-440
45	441-450	21.0	85	10	1010	1.0	2.0	1.0	441-450
46	451-460	21.5	85	10	1010	1.0	2.0	1.0	451-460
47	461-470	22.0	85	10	1010	1.0	2.0	1.0	461-470
48	471-480	22.5	85	10	1010	1.0	2.0	1.0	471-480
49	481-490	23.0	85	10	1010	1.0	2.0	1.0	481-490
50	491-500	23.5	85	10	1010	1.0	2.0	1.0	491-500
51	501-510	24.0	85	10	1010	1.0	2.0	1.0	501-510
52	511-520	24.5	85	10	1010	1.0	2.0	1.0	511-520
53	521-530	25.0	85	10	1010	1.0	2.0	1.0	521-530
54	531-540	25.5	85	10	1010	1.0	2.0	1.0	531-540
55	541-550	26.0	85	10	1010	1.0	2.0	1.0	541-550
56	551-560	26.5	85	10	1010	1.0	2.0	1.0	551-560
57	561-570	27.0	85	10	1010	1.0	2.0	1.0	561-570
58	571-580	27.5	85	10	1010	1.0	2.0	1.0	571-580
59	581-590	28.0	85	10	1010	1.0	2.0	1.0	581-590
60	591-600	28.5	85	10	1010	1.0	2.0	1.0	591-600
61	601-610	29.0	85	10	1010	1.0	2.0	1.0	601-610
62	611-620	29.5	85	10	1010	1.0	2.0	1.0	611-620
63	621-630	30.0	85	10	1010	1.0	2.0	1.0	621-630
64	631-640	30.5	85	10	1010	1.0	2.0	1.0	631-640
65	641-650	31.0	85	10	1010	1.0	2.0	1.0	641-650
66	651-660	31.5	85	10	1010	1.0	2.0	1.0	651-660
67	661-670	32.0	85	10	1010	1.0	2.0	1.0	661-670
68	671-680	32.5	85	10	1010	1.0	2.0	1.0	671-680
69	681-690	33.0	85	10	1010	1.0	2.0	1.0	681-690
70	691-700	33.5	85	10	1010	1.0	2.0	1.0	691-700
71	701-710	34.0	85	10	1010	1.0	2.0	1.0	701-710
72	711-720	34.5	85	10	1010	1.0	2.0	1.0	711-720
73	721-730	35.0	85	10	1010	1.0	2.0	1.0	721-730
74	731-740	35.5	85	10	1010	1.0	2.0	1.0	731-740
75	741-750	36.0	85	10	1010	1.0	2.0	1.0	741-750
76	751-760	36.5	85	10	1010	1.0	2.0	1.0	751-760
77	761-770	37.0	85	10	1010	1.0	2.0	1.0	761-770
78	771-780	37.5	85	10	1010	1.0	2.0	1.0	771-780
79	781-790	38.0	85	10	1010	1.0	2.0	1.0	781-790
80	791-800	38.5	85	10	1010	1.0	2.0	1.0	791-800
81	801-810	39.0	85	10	1010	1.0	2.0	1.0	801-810
82	811-820	39.5	85	10	1010	1.0	2.0	1.0	811-820
83	821-830	40.0	85	10	1010	1.0	2.0	1.0	821-830
84	831-840	40.5	85	10	1010	1.0	2.0	1.0	831-840
85	841-850	41.0	85	10	1010	1.0	2.0	1.0	841-850
86	851-860	41.5	85	10	1010	1.0	2.0	1.0	851-860
87	861-870	42.0	85	10	1010	1.0	2.0	1.0	861-870
88	871-880	42.5	85	10	1010	1.0	2.0	1.0	871-880
89	881-890	43.0	85	10	1010	1.0	2.0	1.0	881-890
90	891-900	43.5	85	10	1010	1.0	2.0	1.0	891-900
91	901-910	44.0	85	10	1010	1.0	2.0	1.0	901-910
92	911-920	44.5	85	10	1010	1.0	2.0	1.0	911-920
93	921-930	45.0	85	10	1010	1.0	2.0	1.0	921-930
94	931-940	45.5	85	10	1010	1.0	2.0	1.0	931-940
95	941-950	46.0	85	10	1010	1.0	2.0	1.0	941-950
96	951-960	46.5	85	10	1010	1.0	2.0	1.0	951-960
97	961-970	47.0	85	10	1010	1.0	2.0	1.0	961-970
98	971-980	47.5	85	10	1010	1.0	2.0	1.0	971-980
99	981-990	48.0	85	10	1010	1.0	2.0	1.0	981-990
100	991-1000	48.5	85	10	1010	1.0	2.0	1.0	991-1000
101	1001-1010	49.0	85	10	1010	1.0	2.0	1.0	1001-1010
102	1011-1020	49.5	85	10	1010	1.0	2.0	1.0	1011-1020
103	1021-1030	50.0	85	10	1010	1.0	2.0	1.0	1021-1030
104	1031-1040	50.5	85	10	1010	1.0	2.0	1.0	1031-1040
105	1041-1050	51.0	85	10	1010	1.0	2.0	1.0	1041-1050
106	1051-1060	51.5	85	10	1010	1.0	2.0	1.0	1051-1060
107	1061-1070	52.0	85	10	1010	1.0	2.0	1.0	1061-1070
108	1071-1080	52.5	85	10	1010	1.0	2.0	1.0	1071-1080
109	1081-1090	53.0	85	10	1010	1.0	2.0	1.0	1081-1090
110	1091-1100	53.5	85	10	1010	1.0	2.0	1.0	1091-1100
111	1101-1110	54.0	85	10	1010	1.0	2.0	1.0	1101-1110
112	1111-1120	54.5	85	10	1010	1.0	2.0	1.0	1111-1120
113	1121-1130	55.0	85	10	1010	1.0	2.0	1.0	1121-1130
114	1131-1140	55.5	85	10	1010	1.0	2.0	1.0	1131-1140
115	1141-1150	56.0	85	10	1010	1.0	2.0	1.0	1141-1150
116	1151-1160	56.5	85	10	1010	1.0	2.0	1.0	1151-1160
117	1161-1170	57.0	85	10	1010	1.0	2.0	1.0	1161-1170
118	1171-1180	57.5	85	10	1010	1.0	2.0	1.0	1171-1180
119	1181-1190	58.0	85	10	1010	1.0	2.0	1.0	1181-1190
120	1191-1200	58.5	85	10	1010	1.0	2.0	1.0	1191-1200
121	1201-1210	59.0	85	10	1010	1.0	2.0	1.0	1201-1210
122	1211-1220	59.5	85	10	1010	1.0	2.0	1.0	1211-1220
123	1221-1230	60.0	85	10	1010	1.0	2.0	1.0	1221-1230
124	1231-1240	60.5	85	10	1010	1.0	2.0	1.0	1231-1240
125	1241-1250	61.0	85	10	1010	1.0	2.0	1.0	1241-1250
126	1251-1260	61.5	85	10	1010	1.0	2.0	1.0	1251-1260
127	1261-1270	62.0	85	10	1010	1.0	2.0	1.0	1261-1270
128	1271-1280	62.5	85	10	1010	1.0	2.0	1.0	1271-1280
129	1281-1290	63.0	85	10	1010	1.0	2.0	1.0	1281-1290</td

VISIBILITY
9.30 KM

TURB LASER MODULE

CALCULATION OF POWER SPECTRUM AND TURBULENCE INDUCED POINTING JITTER OF A LASER TARGET DESIGNATOR AND SEEKER

LASER WAVELENGTH (MICROMETERS)	10.6000
DESIG. APERTURE DIAMETER (METERS)	.101600
BEAMSPREAD ANGLE (RADIANs)	.000130
SEEKER APERTURE DIAMETER (METERS)	.400000
RANGE FROM TARGET TO SEEKER (METERS)	500.00
BEAM SLUE RATE (RAD/SEC)	.000000
DESIGNATION RANGE (METERS)	400.00
DURATION OF TEST (SECONDS)	5.0000
TOTAL DESIGNATOR PATH SEGMENTS	5
TOTAL SEEKER PATH SEGMENTS	5
TOTAL FREQUENCIES FOR WHICH POWER SPECTRUM IS TO BE CALCULATED	512

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN DESIGNATOR PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-012	.93
3	.130000-012	.93
4	.270000-012	.93
5	.500000-013	.93

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN SEEKER PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-012	.93
3	.130000-012	.93
4	.270000-012	.93
5	.500000-013	.93

DESIGNATOR TO TARGET

VIRTUAL POINT SOURCE TO APERTURE DISTANCE	781.533845 (METERS)	
DISTANCE FROM VIRTUAL POINT SOURCE TO TARGET	1181.53845 (METERS)	
INTEGRATED COHERENCE LENGTH	.335957 (METERS)	
DIAMETER/INTEGRATED COHERENCE LENGTH	.302419	
TRANSMITTER-INDUCED BEAM SPREAD	.052000 (METERS)	
DIFFRACTION-LIMITED BEAM SPREAD	.047074 (METERS)	
DIFFRACTION AND TURBULENCE BEAM SPREAD	.047629 (METERS)	
TOTAL EFFECTIVE BEAM SIZE	.172116 (METERS)	
SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY (HERTZ)
1	341660	56.928500
2	5124200	18.976166
3	6129000	11.385700
4	6588533	8.132643
5	13.656743	6.325389

TARGET TO SEEKER

INTEGRATED COHERENCE LENGTH	.934983 (METERS)
DIAMETER/INTEGRATED COHERENCE LENGTH	.184085
TRANSMITTER-INDUCED BEAM SPREAD	.13000-003 (RADIANS)
DIFFRACTION-LIMITED BEAM SPREAD	.065000 (METERS)
DIFFRACTION AND TURBULENCE BEAM SPREAD	.034735 (METERS)
TOTAL EFFECTIVE BEAM SIZE	.0344914 (METERS)
	.2454906 (METERS)

SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY(HERTZ)
1	2.689629	7.400705
2	1.458297	2.460302
3	1.369399	1.489741
4	1.499813	1.952714
5	1.327262	.822301

THE VARIANCE OF THE POWER SPECTRUM IS .8090-011
 OUTPUT FOR DESIGNATOR TO TARGET PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .92021-007, MEAN OF IMAG PART = .00000
 VAR. OF REAL PART = .83284-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = -.52602-011, MEAN OF IMAG PART = .19630-016
 VAR. OF REAL PART = .79425-017, VAR. OF IMAG PART = .13953-031

OUTPUT FOR TARGET TO SEEKER PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .17265-006, MEAN OF IMAG PART = .00000
 VAR. OF REAL PART = .91555-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = .24531-011, MEAN OF IMAG PART = .26733-016
 VAR. OF REAL PART = .87552-017, VAR. OF IMAG PART = .12492-021

XSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE

OPTIONS CHOSEN
MARITIME ARTIC
HORIZONTAL PATH

EXTINCTION FROM 8.0 TO 12.0 MICRONS DISTANCE TRANSMISSION
 KM**-1 KM
 .178 .400 .931+000

OPTIONS CHOSEN
MARITIME POLAR
SLANT PATH FOR 10,600 MICRONS

WARNING FROM SLANT
THE VERTICAL DISTANCE EXCEEDS THE 500 METER UPPER LIMIT, OR
IS NOT AN INTEGER MULTIPLE OF 20 METERS
SLANT DISTANCE CHANGED FROM .2397 TO .2238 KM

SLANT EXTINCTION AT 10.60 MICRONS	DISTANCE	TRANSMISSION	ANGLE
KM**-1	KM		
1.295	.224	.748+000	56.30

OPTIONS CHOSEN
CONTINENTAL POLAR
HORIZONTAL PATH

EXTINCTION FROM 8.0 TO 12.0 MICRONS DISTANCE TRANSMISSION
 KM**-1 KM
 008 .400 .992+000

SMOKE MODEL MODULE

* SMOKE *

EXECUTION 1

WHITE PHOSPHORUS (WP)	METEOROLOGICAL CONDITIONS	EXTINCTION COEFFICIENTS
NO. ROUNDS 1	WIND SPEED 3.6 M/S	0.4-0.7 MICROMETERS .304 M**2/GM
FILL WEIGHT 15,600 LB	WIND DIRECTION (USUAL) 225.0 DEGREES	0.7-1.2 MICROMETERS 2.166 M**2/GM
BURN TIME 100.0 SEC	MET CONVENTION AZIMUTH 225.0 DEGREES	1.06 MICROMETERS 1.541 M**2/GM
EFFICIENCY 100.0 PERCENT	RELATIVE HUMIDITY 87.1 PERCENT	3.0-5.0 MICROMETERS .350 M**2/GM
YIELD FACTOR 5.84	PASQUILL CATEGORY D	8.0-12.0 MICROMETERS .338 M**2/GM
	AIR TEMPERATURE 5.3 DEGREE C	10.6 MICROMETERS .364 M**2/GM
	TEMP. GRADIENT .00 C DEG./M	94.0 GHZ .001 M**2/GM

BURN RATE PROFILE = 1.0000

FIELD COORDINATES				ROTATED COORD.(WIND X-AXIS)				MUNITION ORIGIN	
(X)	(Y)	(Z)	(XW)	(YW)	(ZW)				
MUNITION COORDINATES= -200.00	-50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	METERS
OBSERVER COORDINATES= 200.00	0.00	2.00	2.00	0.00	0.00	2.00	2.00	2.00	METERS
TARGET COORDINATES= -200.00	0.00	2.00	2.00	0.00	0.00	2.00	2.00	2.00	METERS
ANGLE OF ORIGINAL X-AXIS CLOCKWISE WRT NORTH = 90.00 DEG.									
EVENT TIME = .0 SEC									

TIME	LENGTH	WIDTH	HEIGHT	PATHLENGTH	CL	TRANSMISSION			
(SEC)	(METERS)	(METERS)	(METERS)	(METERS)	(GM/M**2)	SPECTRAL BANDS (MICROMETERS)			
0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.0	10.6	94.0 GHZ			
5.	18.	13.	22.	.00	1.000	1.000	1.000	1.000	1.000
10.	36.	16.	34.	.00	1.000	1.000	1.000	1.000	1.000
15.	54.	20.	44.	.00	1.000	1.000	1.000	1.000	1.000
20.	72.	23.	53.	18.16	4.01	1.000	1.000	1.000	1.000
25.	90.	27.	61.	37.92	4.89	1.000	1.000	1.000	1.000
30.	108.	31.	69.	40.38	3.48	1.000	1.000	1.000	1.000
35.	126.	34.	76.	43.87	2.58	1.000	1.000	1.000	1.000
40.	144.	38.	83.	45.86	1.99	1.000	1.000	1.000	1.000
45.	162.	41.	90.	48.25	1.58	1.000	1.000	1.000	1.000
50.	180.	45.	95.	49.49	1.29	1.000	1.000	1.000	1.000
55.	198.	49.	101.	50.64	1.07	1.000	1.000	1.000	1.000
60.	216.	53.	106.	51.64	.90	1.000	1.000	1.000	1.000
65.	234.	56.	111.	52.50	.72	1.000	1.000	1.000	1.000
70.	252.	59.	115.	53.26	.62	1.000	1.000	1.000	1.000
75.	270.	63.	120.	53.90	.55	1.000	1.000	1.000	1.000
80.	288.	67.	125.	54.46	.49	1.000	1.000	1.000	1.000
85.	306.	70.	129.	54.92	.43	1.000	1.000	1.000	1.000
90.	324.	74.	133.	55.30	.37	1.000	1.000	1.000	1.000
95.	342.	77.	136.	55.60	.32	1.000	1.000	1.000	1.000
100.	360.	81.	139.	55.80	.27	1.000	1.000	1.000	1.000
105.	378.	85.	141.	55.94	.23	1.000	1.000	1.000	1.000
110.	396.	88.	144.	56.04	.19	1.000	1.000	1.000	1.000
115.	414.	92.	147.	56.14	.16	1.000	1.000	1.000	1.000
120.	432.	95.	150.	56.60	.12	1.000	1.000	1.000	1.000
125.	450.	99.	153.	56.94	.09	1.000	1.000	1.000	1.000
130.	468.	103.	156.	57.45	.06	1.000	1.000	1.000	1.000
135.	486.	106.	159.	57.93	.04	1.000	1.000	1.000	1.000
140.	504.	110.	160.	58.20	.03	1.000	1.000	1.000	1.000
145.	522.	113.	163.	58.55	.02	1.000	1.000	1.000	1.000
150.	540.	117.	165.	58.87	.01	1.000	1.000	1.000	1.000
155.	558.	121.	167.	59.18	.00	1.000	1.000	1.000	1.000
160.	576.	124.	170.	59.42	-.04	1.000	1.000	1.000	1.000
165.	594.	128.	172.	59.75	-.13	1.000	1.000	1.000	1.000
170.	612.	131.	174.	60.00	-.23	1.000	1.000	1.000	1.000
175.	630.	135.	176.	60.25	-.31	1.000	1.000	1.000	1.000
180.	648.	139.	178.	60.48	-.39	1.000	1.000	1.000	1.000

***TRANSMISSION RETURNED TO MAIN FOR WAVELENGTH OF 10.600 MICRUMETERS IS .999 AT TIME 250.

DIRT TRANSMISSION MODULE

DIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION CALCULATION
*** NOTE -- ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED ***

HT 2.00 PASQUILL CATEGORY D
TEMP 278.36 HT 2.00 WIND 3.60
WIND DIRECTION 45.00

LATITUDE 53.00
THE INVERSION LAYER HEIGHT IS GROWING

SOIL-2
SILT CONTENT .15
DEPTH OF SOD .00

30 DEGREE TILTED TIP AT 0.3 METER DEPTH.
WEIGHT OF CHARGE , 6.80 KG.
DETONATION DEPTH .00

SIMULTANEOUS BURST, UNIFORMLY DISTRIBUTED CHARGES IN A PARALLELOGRAM
TOTAL NUMBER OF CHARGES IS 1, WITH REFERENCE CHARGE AT (.00, .00)
1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (.00, .00)
1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (.00, .00)

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 1.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .767-002

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 19.74 METERS
THE CENTROID COORDINATES ARE 1.34 10.55
THE WIDTH AT THE CENTROID IS 20.62 METERS
THE WIDTH AT 2.00 METERS IS 16.56 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
-5.670 2.000
-2.923 10.550
1.337 19.745
1.337 19.745
11.646 10.550
10.893 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 3.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .620+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 21.89 METERS
THE CENTROID COORDINATES ARE 3.05 10.55
THE WIDTH AT THE CENTROID IS 31.26 METERS
THE WIDTH AT 2.00 METERS IS 11.88 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
-2.947 2.000
-7.608 10.550

9.026 21.938
20.029 20.00000
20.849 10.00000
18.828 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 5.00

WAVELENGTH 10.60 MICRUMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .940+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	.00
THE HEIGHT OF THE CLOUD IS	23.56	METERS
THE CENTROID COORDINATES ARE	18.70	10.50
THE WIDTH AT THE CENTROID IS	38.28	METERS
THE WIDTH AT 2.00 METERS IS	33.37	METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED		
-224	5.000	
-4.436	10.550	
14.704	23.550	
14.704	23.550	
33.844	10.550	
29.151	2.000	

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 7.00

WAVELENGTH 10.60 MICRUMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .987+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	.00
THE HEIGHT OF THE CLOUD IS	24.95	METERS
THE CENTROID COORDINATES ARE	23.39	10.55
THE WIDTH AT THE CENTROID IS	43.78	METERS
THE WIDTH AT 2.00 METERS IS	35.94	METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED		
3.436	5.000	
-1.491	10.550	
21.398	24.948	
21.398	24.948	
43.367	10.550	
39.374	2.000	

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 9.00

WAVELENGTH 10.60 MICRUMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .997+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	.00
THE HEIGHT OF THE CLOUD IS	26.15	METERS
THE CENTROID COORDINATES ARE	28.02	10.55
THE WIDTH AT THE CENTROID IS	48.32	METERS
THE WIDTH AT 2.00 METERS IS	41.56	METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED		
7.097	2.000	

5.906 10 550
5.025 148
5.025 148
4.000 550
4.000 550

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 11.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 27.21 METERS
THE CENTROID COORDINATES ARE 34.76 10.55
THE WIDTH AT THE CENTROID IS 52.30 METERS
THE WIDTH AT 2.00 METERS IS 46.56 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
11.38 2.000
8.605 19.550
34.785 21.130
34.785 21.130
69.349 10.550
57.944 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 13.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 28.17 METERS
THE CENTROID COORDINATES ARE 41.44 10.55
THE WIDTH AT THE CENTROID IS 55.82 METERS
THE WIDTH AT 2.00 METERS IS 50.94 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
15.667 2.000
13.529 19.550
41.439 28.173
41.439 28.173
69.349 10.550
66.605 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 15.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 29.05 METERS
THE CENTROID COORDINATES ARE 49.12 10.55
THE WIDTH AT THE CENTROID IS 59.00 METERS
THE WIDTH AT 2.00 METERS IS 55.00 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

20.265	2.000
19.656	19.550
49.126	29.950
52.126	29.950
75.947	19.550
75.265	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 17.00

WAVELENGTH 10.60 MICRORAMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 29.86 METERS
 THE CENTROID COORDINATES ARE 54.81 10.55
 THE WIDTH AT THE CENTROID IS 61.91 METERS
 THE WIDTH AT 2.00 METERS IS 59.38 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

24.551	2.000
33.850	19.550
34.807	29.959
34.807	29.959
85.783	19.550
85.926	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 19.00

WAVELENGTH 10.60 MICRORAMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 30.61 METERS
 THE CENTROID COORDINATES ARE 61.49 10.55
 THE WIDTH AT THE CENTROID IS 64.60 METERS
 THE WIDTH AT 2.00 METERS IS 63.12 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

29.148	2.000
29.190	19.550
61.490	30.611
61.490	30.611
93.791	19.550
92.273	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 21.00

WAVELENGTH 10.60 MICRORAMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 31.32 METERS
 THE CENTROID COORDINATES ARE 68.17 10.55
 THE WIDTH AT THE CENTROID IS 67.10 METERS
 THE WIDTH AT 2.00 METERS IS 66.87 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

33.434	.00
34.623	10.550
68.174	31.915
68.174	31.915
101.725	10.550
100.309	.00

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 23.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES -200.00	.00	2.00
RECEIVER COORDINATES 200.00	.00	2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001		

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 31.98 METERS
THE CENTROID COORDINATES ARE 74.86 10.55
THE WIDTH AT THE CENTROID IS 69.44 METERS
THE WIDTH AT 2.00 METERS IS 70.31 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

36.032	.00
40.135	10.550
74.858	31.977
74.858	31.977
109.580	10.550
108.344	.00

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 25.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES -200.00	.00	2.00
RECEIVER COORDINATES 200.00	.00	2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001		

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 32.60 METERS
THE CENTROID COORDINATES ARE 81.54 10.55
THE WIDTH AT THE CENTROID IS 71.65 METERS
THE WIDTH AT 2.00 METERS IS 73.75 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

42.630	.00
45.717	10.550
81.541	32.603
81.541	32.603
117.366	10.550
116.380	.00

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 27.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES -200.00	.00	2.00
RECEIVER COORDINATES 200.00	.00	2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001		

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 33.20 METERS
THE CENTROID COORDINATES ARE 68.23 10.55
THE WIDTH AT THE CENTROID IS 73.73 METERS

THE WIDTH AT 2.00 METERS IS 77.19 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
47.2227 10.000
51.5930 10.550
58.1960 10.550
63.1960 10.550
63.1960 10.550
125.691 10.550
124.415 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 29.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT :100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 33.76 METERS
THE CENTROID COORDINATES ARE 94.91 10.55
THE WIDTH AT THE CENTROID IS 75.71 METERS
THE WIDTH AT 2.00 METERS IS 80.94 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
81.5129 2.000
97.6509 10.550
94.909 33.761
94.909 33.761
132.763 10.550
132.450 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 31.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT :100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 34.30 METERS
THE CENTROID COORDINATES ARE 101.59 10.55
THE WIDTH AT THE CENTROID IS 77.59 METERS
THE WIDTH AT 2.00 METERS IS 84.06 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
56.111 2.000
62.799 10.550
101.593 34.299
101.593 34.299
140.387 10.550
140.173 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 33.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT :100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 34.81 METERS
THE CENTROID COORDINATES ARE 108.28 10.55

THE WIDTH AT THE CENTROID IS 79.38 METERS
THE WIDTH AT 2.00 METERS IS 87.19 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

60.706	2.000
68.585	10.550
108.276	34.814
108.276	34.814
147.968	10.550
147.968	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 35.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT :100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 35.31 METERS
THE CENTROID COORDINATES ARE 114.96 10.55
THE WIDTH AT THE CENTROID IS 81.10 METERS
THE WIDTH AT 2.00 METERS IS 90.31 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

65.306	2.000
74.410	10.550
114.960	35.308
114.960	35.308
155.510	10.550
155.619	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 37.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT :100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 35.78 METERS
THE CENTROID COORDINATES ARE 121.64 10.55
THE WIDTH AT THE CENTROID IS 92.75 METERS
THE WIDTH AT 2.00 METERS IS 93.75 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

69.592	2.000
80.271	10.550
121.644	35.782
121.644	35.782
163.017	10.550
163.342	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 39.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT :100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 36.24 METERS

THE CENTROID COORDINATES ARE 128.33 10.55
THE WIDTH AT THE CENTROID IS .84.33 METERS
THE WIDTH AT 2.00 METERS IS .96.88 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
74.190 2.000
86.164 10.550
128.328 36.238
128.328 36.238
170.492 10.550
171.065 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 41.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 36.68 METERS
THE CENTROID COORDINATES ARE 135.01 10.55
THE WIDTH AT THE CENTROID IS .85.85 METERS
THE WIDTH AT 2.00 METERS IS .100.00 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
78.787 2.000
92.086 10.550
135.011 36.677
135.011 36.677
177.936 10.550
178.787 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 43.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 37.10 METERS
THE CENTROID COORDINATES ARE 141.70 10.55
THE WIDTH AT THE CENTROID IS .87.32 METERS
THE WIDTH AT 2.00 METERS IS .102.81 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
83.385 2.000
98.037 10.550
141.695 37.102
141.695 37.102
185.353 10.550
186.198 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 45.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00

THE HEIGHT OF THE CLOUD IS 37.51 METERS
THE CENTROID COORDINATES ARE 148.379 10.55
THE WIDTH AT THE CENTROID IS 88.798 METERS
THE WIDTH AT 2.00 METERS IS 109.94 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
87.983 2.000
104.013 10.550
148.379 37.512
148.379 37.512
192.745 10.550
193.921 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 47.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 37.91 METERS
THE CENTROID COORDINATES ARE 155.06 10.55
THE WIDTH AT THE CENTROID IS 90.10 METERS
THE WIDTH AT 2.00 METERS IS 108.75 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
92.581 2.000
110.012 10.550
155.063 37.909
155.063 37.909
200.113 10.550
201.331 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 49.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 38.29 METERS
THE CENTROID COORDINATES ARE 161.75 10.55
THE WIDTH AT THE CENTROID IS 81.43 METERS
THE WIDTH AT 2.00 METERS IS 111.87 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
97.179 2.000
116.034 10.550
161.746 38.293
161.746 38.293
207.459 10.550
209.054 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 51.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 38.67 METERS
 THE CENTROID COORDINATES ARE 168.43 10.55
 THE WIDTH AT THE CENTROID IS 92.71 METERS
 THE WIDTH AT 2.00 METERS IS 114.69 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 101.777 2.000
 122.076 10.550
 168.430 38.665
 168.430 38.665
 214.784 10.550
 216.464 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 53.00

WAVELENGTH 10.60 MICRORAMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 39.03 METERS
 THE CENTROID COORDINATES ARE 175.11 10.55
 THE WIDTH AT THE CENTROID IS 93.95 METERS
 THE WIDTH AT 2.00 METERS IS 117.50 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 106.375 2.000
 128.138 10.550
 175.114 39.027
 175.114 39.027
 222.090 10.550
 223.875 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 55.00

WAVELENGTH 10.60 MICRORAMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 39.38 METERS
 THE CENTROID COORDINATES ARE 181.80 10.55
 THE WIDTH AT THE CENTROID IS 95.16 METERS
 THE WIDTH AT 2.00 METERS IS 120.00 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 111.285 2.000
 134.217 10.550
 181.797 39.378
 181.797 39.378
 229.377 10.550
 231.285 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 57.00

WAVELENGTH 10.60 MICRORAMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 39.72 METERS
THE CENTROID COORDINATES ARE 188.48 10.55
THE WIDTH AT THE CENTROID IS 96.33 METERS
THE WIDTH AT 2.00 METERS IS 122.81 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
115.883 2.000
140.315 10.550
168.481 39.720
168.481 39.720
236.648 10.550
238.695 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 59.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 40.05 METERS
THE CENTROID COORDINATES ARE 195.16 10.55
THE WIDTH AT THE CENTROID IS 97.47 METERS
THE WIDTH AT 2.00 METERS IS 125.62 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
120.481 2.000
146.428 10.550
195.165 40.052
195.165 40.052
243.902 10.550
246.106 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 61.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 40.38 METERS
THE CENTROID COORDINATES ARE 201.85 10.55
THE WIDTH AT THE CENTROID IS 98.58 METERS
THE WIDTH AT 2.00 METERS IS 128.12 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
125.391 2.000
152.557 10.550
201.849 40.376
201.849 40.376
251.140 10.550
253.516 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 63.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 40.69 METERS
THE CENTROID COORDINATES ARE 208.53 10.55
THE WIDTH AT THE CENTROID IS 99.66 METERS
THE WIDTH AT 2.00 METERS IS 130.62 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
130.301 2.000
158.700 10.550
208.532 40.691
208.532 40.691
258.364 10.550
260.326 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 65.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 41.00 METERS
THE CENTROID COORDINATES ARE 215.22 10.55
THE WIDTH AT THE CENTROID IS 100.72 METERS
THE WIDTH AT 2.00 METERS IS 133.12 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
135.212 2.000
164.857 10.550
215.216 40.998
215.216 40.998
265.875 10.550
268.337 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 67.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 41.30 METERS
THE CENTROID COORDINATES ARE 221.90 10.55
THE WIDTH AT THE CENTROID IS 101.74 METERS
THE WIDTH AT 2.00 METERS IS 135.00 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
140.435 2.000
171.028 10.550
221.900 41.098
221.900 41.098
262.772 10.550
275.435 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 69.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 41.59 METERS
THE CENTROID COORDINATES ARE 228.58 10.55
THE WIDTH AT THE CENTROID IS 103.75 METERS
THE WIDTH AT 2.00 METERS IS 137.50 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
145.345 2.000
177.211 10.550
228.584 41.591
228.584 41.591
279.956 10.550
282.845 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 71.00

WAVELENGTH 10.60 MICRORAMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 41.88 METERS
THE CENTROID COORDINATES ARE 235.27 10.55
THE WIDTH AT THE CENTROID IS 103.72 METERS
THE WIDTH AT 2.00 METERS IS 139.69 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
150.568 2.000
183.406 10.550
235.267 41.877
235.267 41.877
287.129 10.550
290.255 2.000

LASER TRANSMITTANCE MODULE

WAVELENGTH (MICRONS)	H ₂ O PRESSURE (TORR)	TEMPERATURE (HRS)	ABSORPTION COEFFICIENT (KM ⁻¹)	LINE	PATH LENGTH (KM)	TRANSMISSION
10.591	5.821	CHANGED TO 10.591 NEAREST STANDARD WAVELENGTH *** P<20,	.111+000	.278.46	.4000+000	.9565+000

*** WARNING INPUT WAVELENGTH 10.600 CHANGED TO 10.591 NEAREST STANDARD WAVELENGTH *** P<20,

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .719 KM

TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .481 KM

TOTAL OPTICAL DEPTH = 25.11

TRANSMITTANCE ALONG LINE-OF-SIGHT = 12449-010

SEEKER COORDINATES (KM)	TARGET COORDINATES (KM)
XSEEKER	XTARGET
ZSEEKER	ZTARGET
.200	.000
.600	-.260
	.000
	.002

CLOUD TYPE	LINE-OF-SIGHT INTERSECTION COORDINATES (KM)			
ID NUMBER	XUPPER	ZUPPER	XLOWER	ZLOWER
ST/ 1	.200	.000	.600	-.068
				.000
				.200

CLOUD TYPE	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ID NUMBER					
ST/ 1	.200	.400	.000	25.11	12449-010

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .599 KM

TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .361 KM

TOTAL OPTICAL DEPTH = 16.57

TRANSMITTANCE ALONG LINE-OF-SIGHT = .63594-007

SEEKER COORDINATES (KM)	TARGET COORDINATES (KM)
XSEEKER	XTARGET
YSEEKER	YTARGET
ZSEEKER	ZTARGET

.133 .000 .500 -.200 .000 .002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT RUMBLE	INTERSECTION ZIPPER	COORDINATES XLOWER	COORDINATES YLOWER	COORDINATES ZLOWER
ST/ 1	.133 .000	.500	-.068	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	16.57	.63554-007

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .359 KM
TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .120 KM

TOTAL OPTICAL DEPTH = 3.02

TRANSMITTANCE ALONG LINE-OF-SIGHT = .48846-001

SEEKER COORDINATES (KM) XSEEKER	SEEKER COORDINATES (KM) YSEEKER	SEEKER COORDINATES (KM) ZSEEKER	TARGET COORDINATES (KM) XTARGET	TARGET COORDINATES (KM) YTARGET	TARGET COORDINATES (KM) ZTARGET
.000	.000	.300	-.200	.000	.002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT XBUFFER	LINE-OF-SIGHT YBUFFER	LINE-OF-SIGHT ZBUFFER	INTERSECTION XLOWER	INTERSECTION YLOWER	INTERSECTION ZLOWER	COORDINATES (KM) XLOWER	COORDINATES (KM) YLOWER	COORDINATES (KM) ZLOWER
ST/ 1	.000	.000	.000	.300	-.067	.000	.200	.200	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	3.02	.48846-001

CWIC MUNITION EXPENDITURES / INVERSE STATIC TARGET DETECTION MODULE

DEVICE NUMBER	8	TARGET INTRINSIC CONTRAST OF TEMPERATURE DIFFERENCE				1.000
FOV TYPE -	WIDE	MINIMUM TARGET DIMENSION (M)				2.300
FOV < DEG >	10.620	ACQUISITION LEVEL (50 PCNT)				1.000
MAGNIFICATION	1.000 (ARBITRARY DEFAULT)	SEARCH ZONE (DEGREES**2)				81.000
FOR NO GREATER INPUT DETECTION PROB.	UNDER INPUT DETECTION TIME(SEC)	AND INPUT SKY/GROUND RATIO	AT COMPUTED RESOLVE CYCLES, RC	COMPUTED CONTRAST OR TEMP OFF. TOTAL PATH DEVICE	COMPUTED TEMP DIFF.	COMMENTS
.990	1.32	10.000000	2.000	.24	.773	.037
.990	1.58	10.000000	2.000	.24	.567	.037
.990	2.16	10.000000	2.000	.24	.446	.037
.750	2.16	10.000000	2.000	.24	.313	.037
.500	3.89	10.000000	2.000	.24	.313	.037

*** CONTINUED ON NEXT PAGE ***

```

*****  

* INVERSE STATIC TARGET DETECTION MODEL *  

*****  

*****  

DEVICE NUMBER 8  

FOV TYPE - WIDE  

FOV <DEG> 10.620  

MAGNIFICATION 1.000 (ARBITRARY DEFAULT)  

FOR NO GREATER  

DETECTION PROB.  

TIME SEC>  

.100 24.64  

*****  

TARGET INTRINSIC CONTRAST OR  

TEMPERATURE DIFFERENCE 1.000  

MINIMUM TARGET DIMENSION (M) 2.300  

ACQUISITION LEVEL (50 PCNT) 1.000  

SEARCH ZONE <DEGREES>*2> 81.000  

REQUIRES <TO DEFEAT DEVICE AT MOST  

COMPUTED CONTRAST OR COMPUTED  

RESOLVABLE RANGE TEMP OFF, TOTAL PATH  

CYCLES, SEC AT DEVICE TRANSMITTANCE  

.037  

.037  

.059  

.24  

.2000  

10.000000  

.100  

*****  

NOTE: INPUT DETECTION PROBAB-  

ILITY REQUIRES THAT (CURRENT  

TEMP DIFF.) BELOW THRESHOLD  

VALUES ASSUMED ARE 39 PERCENT  

OF THRESHOLD. ADDITIONAL OBS-  

CURRENT WILL NOT DECREASE  

DETECTION PROBABILITY.

```

*** FINAL TOTAL TRANSMISSION FROM ITAM = .037

* CWIC MUNITION EXPENDITURES *
* *****

MUNITION EXPENDITURES
FOR HC AND WP SMOKE

SLANT RANGE OBS-TGT	- KM	= .400
AZIMUTH OF TARGET	- DEG	= 000
AVG ROUGHNESS ELEMENT	- CM	= 74.6
ATMOSPHERIC EXTINCTION CORRECTIONS		
CORRECTED FOR VISIBILITY	-	YES
MARITIME ARCTIC AIR MASS	-	NO
MARITIME POLAR AIR MASS	-	NO
CONTINENTAL POLAR AIR MASS	-	NO
CORRECTED FOR RAIN	-	NO
CORRECTED FOR SNOW	-	NO
METEOROLOGICAL INPUTS		
WIND SPEED	- M/SEC	= 3.60
WIND DIRECTION	- DEG	= 225.00
PASQUILL CATEGORY	-	
VISIBILITY	- KM	= 9.302
RELATIVE HUMIDITY	- PERCENT	= 87.1
TRANSMISSION THRESHOLDS		
VISIBLE:	TOTAL	SMOKE
NEAR IR:	.037	.043
MID IR:	.037	.044
FAR IR:	.037	.039

- VISIBLE:		- NEAR IR:	
		DURATION MINUTES	DURATION MINUTES
SCREEN	LENGTH METERS 400.	SCREEN	HC SMOKE SCREEN
		15.00	400.
	HC SMOKE SCREEN		
VOLLEY	105MM HOWITZER		105MM HOWITZER
	GUNS RATE SPACING ROUNDS		GUNS RATE SPACING ROUNDS
INITIAL:	.8.	INITIAL:	.8.
SUSTAINING:	.5	SUSTAINING:	.5
	51.		18.
	51.		18.
	60.		173.
VOLLEY	155MM HOWITZER		155MM HOWITZER
	GUNS RATE SPACING ROUNDS		GUNS RATE SPACING ROUNDS
INITIAL:	.3.	INITIAL:	.4.
SUSTAINING:	.2.	SUSTAINING:	.4.
	162.		114.
	328.		114.
	16.		30.
	UP SMOKE SCREEN		UP SMOKE SCREEN
VOLLEY	105MM HOWITZER		105MM HOWITZER
	GUNS RATE SPACING ROUNDS		GUNS RATE SPACING ROUNDS
INITIAL:	.6.	INITIAL:	.8.
SUSTAINING:	.6.	SUSTAINING:	.8.
	.82.		.82.
	.82.		.82.
	129.		129.
VOLLEY	155MM HOWITZER		155MM HOWITZER
	GUNS RATE SPACING ROUNDS		GUNS RATE SPACING ROUNDS
INITIAL:	.3.	INITIAL:	.4.
SUSTAINING:	.2.	SUSTAINING:	.3.
	.8		.8
	248.		196.
	248.		196.
	31.		41.

- MID IR: -

SCREEN	LENGTH METERS 400.	DURATION MINUTES 15.00	
WP SMOKE SCREEN			
	ROUNDS/ 60 METERS	RATE/ MINUTE	TOTAL ROUNDS
105MM:	9.	1.	1312.
155MM:	3.	1.	495.

- FAR IR: -

SCREEN	LENGTH METERS 400.	DURATION MINUTES 15.00	
WP SMOKE SCREEN			
	ROUNDS/ 60 METERS	RATE/ MINUTE	TOTAL ROUNDS
105MM:	9.	1.	989.
155MM:	2.	1.	330.

OVERCAST SKY RADIATIVE TRANSFER MODULE

```
-- RADIATION UNDER OVERCAST SKY --
X0    =    - .067 <KM>      XT    =    - .200 <KM>
Y0    =    -.000                  YT    =    -.000
Z0    =    .200                  ZT    =    .000
LAMBDA =    10.600 <NMU>      LC    =    2.0000+002 <W/M2-SR-MU>
TEMP   =    19.800 <DEG C>     LG    =    5.0000+001
KAPPA  =    4.0000-001 <1/KM>   LB0   =    5.0000+001
ETA    =    .700                 L0    =    1.000

**THERMAL CALCULATION OF PATH RADIANCE
BBTEMP= 7.4082+000 W/M2-SR-MU

PATH LENGTH <KM>  TRANSMITTANCE  PATH RADIANCE  CONTRAST
+-----+ +-----+ +-----+
.239          .00007       2.409+001      .00015

-- RADIATION UNDER OVERCAST SKY --
X0    =    - .133 <KM>      XT    =    - .200 <KM>
Y0    =    -.000                  YT    =    -.000
Z0    =    .000                  ZT    =    .000
LAMBDA =    10.600 <NMU>      LC    =    2.0000+002 <W/M2-SR-MU>
TEMP   =    19.800 <DEG C>     LG    =    5.0000+001
KAPPA  =    4.0000-001 <1/KM>   LB0   =    5.0000+001
ETA    =    .700                 L0    =    1.000

**THERMAL CALCULATION OF PATH RADIANCE
BBTEMP= 7.4082+000 W/M2-SR-MU

PATH LENGTH <KM>  TRANSMITTANCE  PATH RADIANCE  CONTRAST
+-----+ +-----+ +-----+
.119          .00872       9.826-001      .30742
```

SELF-SCREENING SMOKE GRENADE MODULE

```
*****  
***** PROGFM GRNADE  
***** EUSAEEL80  
*****
```

*****CARD INPUT*****

NAME							
OUTP	.000	.000	.000	.000	.000	.000	.000
DBSC	200.000	.000	2.000	.000	.000	.000	.000
MUNC	-200.000	.000	2.000	95.000	100.000	10.000	.000
TARC	-200.000	40.000	2.000	.000	.000	.000	.000
BART	5.000	400.000	5.000	90.000	.000	.000	.000
MINT	1.000	.793	14.300	1.000	.000	4.700	.070
HETR	50.000	2.000	220.000	4.000	20.000	.000	.000
EXTC	.000	.000	.000	.000	.000	.000	.000
BURN	.000	.000	.000	.000	.000	.000	.000
MISC	.000	.000	.000	.000	.000	.000	.000
GO	.000	.000	.000	.000	.000	.000	.000

*****INPUT***** ALL LENGTHS IN METERS

METEOROLOGICAL:

WIND SPEED 3.6 M/S
WIND DIRECTION 225.0 DEG

PASQUILL CATEGORY 4

RELATIVE HUMIDITY 87.1 % CLOCKWISE FROM NORTH (DCWFN)

NOTE: X AXIS HEADING: 87.1 DEG EXTINCTION COEFFICIENTS:

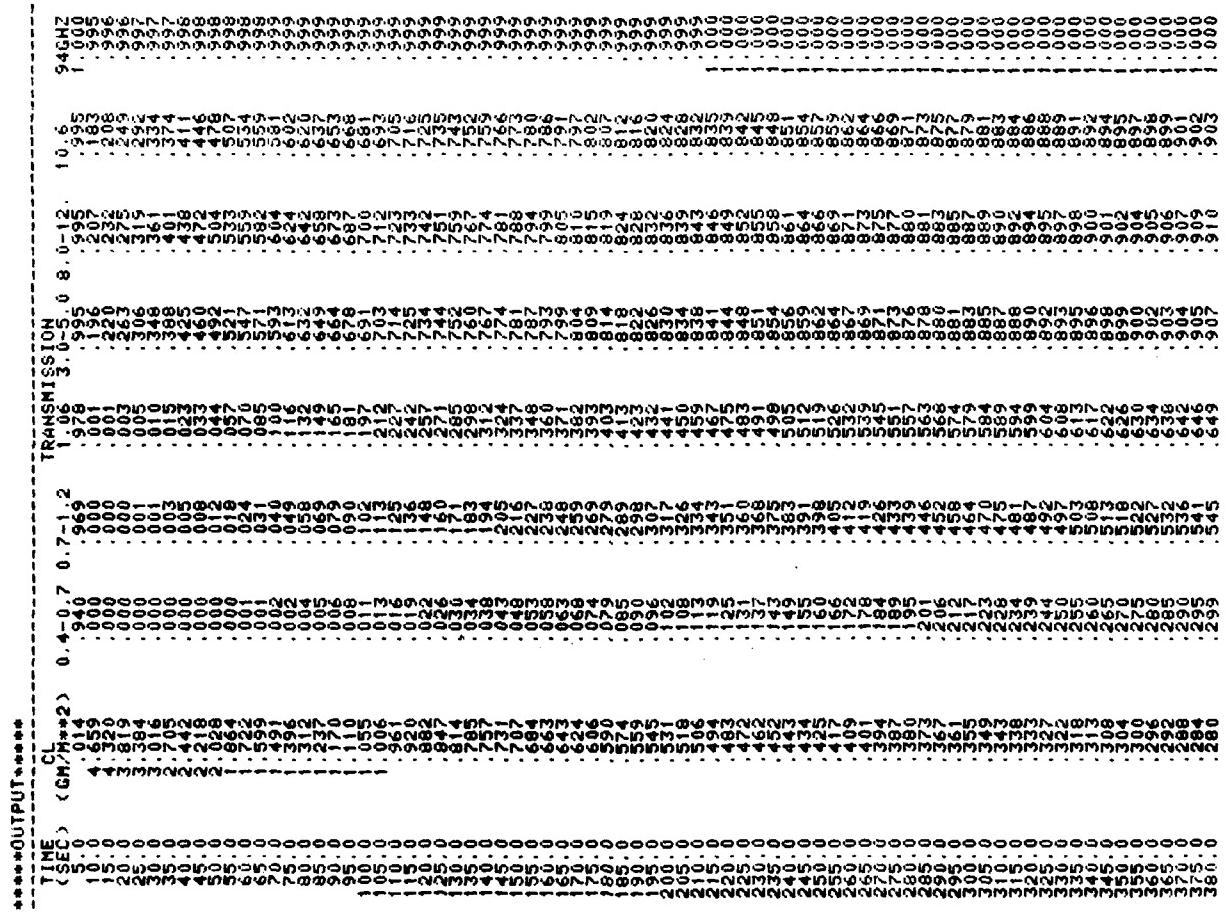
WINDS 1.0000000000000000 M**2/GM

OBSERVER/TARGET:

X(OBS)	200.0	0	0	0	0	0	0
Y(OBS)	2.0	0	0	0	0	0	0
Z(OBS)	2.0	0	0	0	0	0	0
X(TAR)	-200.0	3.0	0	0	0	0	0
Y(TAR)	-40.0	0	0	0	0	0	0
Z(TAR)	2.0	0	0	0	0	0	0

TANK/MUNITION DATA:

X(TANK)	-200.0	0	0	SIGZ(XREF)	7.2
Y(TANK)	0	0	0	XREF	
Z(TANK)	2.0	0	0	MIXING HEIGHT(M), SCAVENGING COEFF(HK)	100.0
HEADING(DCWFN)	95.0	0	0	SCAVENGING COEFF(ERC)	300.0
RANGE(GM)	100.0	0	0	SCAVENGING VELOCITY(VS)	1.000
NO GRENADES	100.0	0	0	SUPERVISORIAL DIFF EXPONENT(SUPOUR)	0.21
SMOKE THICKNESS(GM)	350.0	0	0	VERTICAL DIFF EXPONENT(ZDIFF)	1.455
LINE LENGTH	10.0	0	0	CROSSWIND DIFF CONSTANT(YDIFF)	4.7
BURN CONSTANT	.070	1/S	0	YIELD FACTOR	
EFFICIENCY	62.0	0	0		



385.0	.277	.304	.549	.653	.908	.911	.904	.000
390.0	.273	.309	.554	.657	.909	.912	.905	.000
395.0	.270	.313	.558	.660	.910	.913	.907	.000
400.0	.266	.318	.562	.664	.911	.914	.908	.000

*****CARD INPUT*****

DONE PROGRAM GRNADE END*****

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .2331-019

RUN NUMBER 2

INDIVIDUAL MODULES	SELECTED FCLOUD
WAVENUMBER(CM**-1)	BEGINNING 9433.962
WAVELENGTH(MICRONS)	ENDING 9433.962
FREQUENCY(GHZ)	1.060 283018.863

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

CLASS 1	= FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS 2	= FOG, HAZE AND MIST WITH VIS LT 3 KM.
CLASS 3	= HAZE AND MIST WITH VIS GE 7 KM.
CLASS 4	= VIS GE 3 KM.
CLASS 5	= VIS GE 1 KM.
CLASS 6	= VIS GE .3 KM.
CLASS 7	= VIS LT 1 KM.
CLASS 8	= VIS LT 3 KM.
CLASS 9	= VIS GE 7 KM.
CLASS 10	= RAIN AND TSMS WITH VIS LT 1 KM.
CLASS 11	= RAIN AND TSMS WITH VIS LT 3 KM.
CLASS 12	= RAIN AND TSMS WITH VIS GE 7 KM.
CLASS 13	= DRIZZLE, RAIN AND TSMS WITH VIS LT 1 KM.
CLASS 14	= DRIZZLE, RAIN AND TSMS WITH VIS LT 3 KM.
CLASS 15	= DRIZZLE, RAIN AND TSMS WITH VIS GE 7 KM.
CLASS 16	= SNOW, SLEET AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 17	= SNOW, SLEET AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 18	= SNOW, SLEET AND ABSOLUTE HUMIDITY LT 300 M.
CLASS 19	= SNOW, SLEET AND ABSOLUTE HUMIDITY GE 300 M.
CLASS 20	= CEILING HEIGHT LT 300 M.
CLASS 21	= CEILING HEIGHT LT 1000 M.
CLASS 22	= NO CEILING HEIGHT.
	= ALL CONDITIONS COMBINED.

EDSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS

DURING SPRING AT 03-09 (LST).

६३६

CLASS NO.	FREQUENCY	WIND DIRECTION	WIND CLASS	VISIBILITY	
				1	2
1	179	W. N.W.	<25	10-15	10-15
2	40	S.E.	25-35	10-15	10-15
3	1	N.E.	35-45	10-15	10-15
4	1	E.	45-55	10-15	10-15
5	1	S.E.	55-65	10-15	10-15
6	1	S.	65-75	10-15	10-15
7	1	S.W.	75-85	10-15	10-15
8	1	W.S.W.	85-95	10-15	10-15
9	1	W.	95-105	10-15	10-15
10	1	W.N.W.	105-115	10-15	10-15
11	1	N.W.	115-125	10-15	10-15
12	1	N.	125-135	10-15	10-15
13	1	N.E.	135-145	10-15	10-15
14	1	E.N.E.	145-155	10-15	10-15
15	1	E.	155-165	10-15	10-15
16	1	E.N.E.	165-175	10-15	10-15
17	1	N.E.	175-185	10-15	10-15
18	1	N.	185-195	10-15	10-15
19	1	N.W.	195-205	10-15	10-15
20	1	W.	205-215	10-15	10-15
21	1	W.N.W.	215-225	10-15	10-15
22	1	N.W.	225-235	10-15	10-15
23	1	N.	235-245	10-15	10-15
24	1	N.E.	245-255	10-15	10-15
25	1	E.N.E.	255-265	10-15	10-15
26	1	E.	265-275	10-15	10-15
27	1	E.N.E.	275-285	10-15	10-15
28	1	N.E.	285-295	10-15	10-15
29	1	N.	295-305	10-15	10-15
30	1	N.W.	305-315	10-15	10-15
31	1	W.	315-325	10-15	10-15
32	1	W.N.W.	325-335	10-15	10-15
33	1	N.W.	335-345	10-15	10-15
34	1	N.	345-355	10-15	10-15
35	1	N.E.	355-365	10-15	10-15
36	1	E.N.E.	365-375	10-15	10-15
37	1	E.	375-385	10-15	10-15
38	1	E.N.E.	385-395	10-15	10-15
39	1	N.E.	395-405	10-15	10-15
40	1	N.	405-415	10-15	10-15
41	1	N.W.	415-425	10-15	10-15
42	1	W.	425-435	10-15	10-15
43	1	W.N.W.	435-445	10-15	10-15
44	1	N.W.	445-455	10-15	10-15
45	1	N.	455-465	10-15	10-15
46	1	N.E.	465-475	10-15	10-15
47	1	E.N.E.	475-485	10-15	10-15
48	1	E.	485-495	10-15	10-15
49	1	E.N.E.	495-505	10-15	10-15
50	1	N.E.	505-515	10-15	10-15
51	1	N.	515-525	10-15	10-15
52	1	N.W.	525-535	10-15	10-15
53	1	W.	535-545	10-15	10-15
54	1	W.N.W.	545-555	10-15	10-15
55	1	N.W.	555-565	10-15	10-15
56	1	N.	565-575	10-15	10-15
57	1	N.E.	575-585	10-15	10-15
58	1	E.N.E.	585-595	10-15	10-15
59	1	E.	595-605	10-15	10-15
60	1	E.N.E.	605-615	10-15	10-15
61	1	N.E.	615-625	10-15	10-15
62	1	N.	625-635	10-15	10-15
63	1	N.W.	635-645	10-15	10-15
64	1	W.	645-655	10-15	10-15
65	1	W.N.W.	655-665	10-15	10-15
66	1	N.W.	665-675	10-15	10-15
67	1	N.	675-685	10-15	10-15
68	1	N.E.	685-695	10-15	10-15
69	1	E.N.E.	695-705	10-15	10-15
70	1	E.	705-715	10-15	10-15
71	1	E.N.E.	715-725	10-15	10-15
72	1	N.E.	725-735	10-15	10-15
73	1	N.	735-745	10-15	10-15
74	1	N.W.	745-755	10-15	10-15
75	1	W.	755-765	10-15	10-15
76	1	W.N.W.	765-775	10-15	10-15
77	1	N.W.	775-785	10-15	10-15
78	1	N.	785-795	10-15	10-15
79	1	N.E.	795-805	10-15	10-15
80	1	E.N.E.	805-815	10-15	10-15
81	1	E.	815-825	10-15	10-15
82	1	E.N.E.	825-835	10-15	10-15
83	1	N.E.	835-845	10-15	10-15
84	1	N.	845-855	10-15	10-15
85	1	N.W.	855-865	10-15	10-15
86	1	W.	865-875	10-15	10-15
87	1	W.N.W.	875-885	10-15	10-15
88	1	N.W.	885-895	10-15	10-15
89	1	N.	895-905	10-15	10-15
90	1	N.E.	905-915	10-15	10-15
91	1	E.N.E.	915-925	10-15	10-15
92	1	E.	925-935	10-15	10-15
93	1	E.N.E.	935-945	10-15	10-15
94	1	N.E.	945-955	10-15	10-15
95	1	N.	955-965	10-15	10-15
96	1	N.W.	965-975	10-15	10-15
97	1	W.	975-985	10-15	10-15
98	1	W.N.W.	985-995	10-15	10-15
99	1	N.W.	995-1005	10-15	10-15
100	1	N.	1005-1015	10-15	10-15
101	1	N.E.	1015-1025	10-15	10-15
102	1	E.N.E.	1025-1035	10-15	10-15
103	1	E.	1035-1045	10-15	10-15
104	1	E.N.E.	1045-1055	10-15	10-15
105	1	N.E.	1055-1065	10-15	10-15
106	1	N.	1065-1075	10-15	10-15
107	1	N.W.	1075-1085	10-15	10-15
108	1	W.	1085-1095	10-15	10-15
109	1	W.N.W.	1095-1105	10-15	10-15
110	1	N.W.	1105-1115	10-15	10-15
111	1	N.	1115-1125	10-15	10-15
112	1	N.E.	1125-1135	10-15	10-15
113	1	E.N.E.	1135-1145	10-15	10-15
114	1	E.	1145-1155	10-15	10-15
115	1	E.N.E.	1155-1165	10-15	10-15
116	1	N.E.	1165-1175	10-15	10-15
117	1	N.	1175-1185	10-15	10-15
118	1	N.W.	1185-1195	10-15	10-15
119	1	W.	1195-1205	10-15	10-15
120	1	W.N.W.	1205-1215	10-15	10-15
121	1	N.W.	1215-1225	10-15	10-15
122	1	N.	1225-1235	10-15	10-15
123	1	N.E.	1235-1245	10-15	10-15
124	1	E.N.E.	1245-1255	10-15	10-15
125	1	E.	1255-1265	10-15	10-15
126	1	E.N.E.	1265-1275	10-15	10-15
127	1	N.E.	1275-1285	10-15	10-15
128	1	N.	1285-1295	10-15	10-15
129	1	N.W.	1295-1305	10-15	10-15
130	1	W.	1305-1315	10-15	10-15
131	1	W.N.W.	1315-1325	10-15	10-15
132	1	N.W.	1325-1335	10-15	10-15
133	1	N.	1335-1345	10-15	10-15
134	1	N.E.	1345-1355	10-15	10-15
135	1	E.N.E.	1355-1365	10-15	10-15
136	1	E.	1365-1375	10-15	10-15
137	1	E.N.E.	1375-1385	10-15	10-15
138	1	N.E.	1385-1395	10-15	10-15
139	1	N.	1395-1405	10-15	10-15
140	1	N.W.	1405-1415	10-15	10-15
141	1	W.	1415-1425	10-15	10-15
142	1	W.N.W.	1425-1435	10-15	10-15
143	1	N.W.	1435-1445	10-15	10-15
144	1	N.	1445-1455	10-15	10-15
145	1	N.E.	1455-1465	10-15	10-15
146	1	E.N.E.	1465-1475	10-15	10-15
147	1	E.	1475-1485	10-15	10-15
148	1	E.N.E.	1485-1495	10-15	10-15
149	1	N.E.	1495-1505	10-15	10-15
150	1	N.	1505-1515	10-15	10-15
151	1	N.W.	1515-1525	10-15	10-15
152	1	W.	1525-1535	10-15	10-15
153	1	W.N.W.	1535-1545	10-15	10-15
154	1	N.W.	1545-1555	10-15	10-15
155	1	N.	1555-1565	10-15	10-15
156	1	N.E.	1565-1575	10-15	10-15
157	1	E.N.E.	1575-1585	10-15	10-15
158	1	E.	1585-1595	10-15	10-15
159	1	E.N.E.	1595-1605	10-15	10-15
160	1	N.E.	1605-1615	10-15	10-15
161	1	N.	1615-1625	10-15	10-15
162	1	N.W.	1625-1635	10-15	10-15
163	1	W.	1635-1645	10-15	10-15
164	1	W.N.W.	1645-1655	10-15	10-15
165	1	N.W.	1655-1665	10-15	10-15
166	1	N.	1665-1675	10-15	10-15
167	1	N.E.	1675-1685	10-15	10-15
168	1	E.N.E.	1685-1695	10-15	10-15
169	1	E.	1695-1705	10-15	10-15
170	1	E.N.E.	1705-1715	10-15	10-15
171	1	N.E.	1715-1725	10-15	10-15
172	1	N.	1725-1735	10-15	10-15
173	1	N.W.	1735-1745	10-15	10-15
174	1	W.	1745-1755	10-15	10-15
175	1	W.N.W.	1755-1765	10-15	10-15
176	1	N.W.	1765-1775	10-15	10-15
177	1	N.	1775-1785	10-15	10-15
178	1	N.E.	1785-1795	10-15	10-15
179	1	E.N.E.	1795-1805	10-15	10-15
180	1	E.	1805-1815	10-15	10-15
181	1	E.N.E.	1815-1825	10-15	10-15
182	1	N.E.	1825-1835	10-15	10-15
183	1	N.	1835-1845	10-15	10-15
184	1	N.W.	1845-1855	10-15	10-15
185	1	W.	1855-1865	10-15	10-15
186	1	W.N.W.	1865-1875	10-15	10-15
187	1	N.W.	1875-1885	10-15	10-15
188	1	N.	1885-1895	10-15	10-15
189	1	N.E.	1895-1905	10-15	10-15
190	1	E.N.E.	1905-1915	10-15	10-15
191	1	E.	1915-1925	10-15	10-15
192	1	E.N.E.	1925-1935	10-15	10-15
193	1	N.E.	1935-1945	10-15	10-15
194	1	N.	1945-1955	10-15	10-15
195	1	N.W.	1955-1965	10-15	10-15
196	1	W.	1965-1975	10-15	10-15
197	1	W.N.W.	1975-1985	10-15	10-15
198	1	N.W.	1985-1995	10-15	10-15
199	1	N.	1995-2005	10-15	10-15
200	1	N.E.	2005-2015	10-15	10-15
201	1	E.N.E.	2015-2025	10-15	10-15
202	1	E.	2025-2035	10-15	10-15
203	1	E.N.E.	2035-2045	10-15	10-15
204	1	N.E.	2045-2055	10-15	10-15
205	1	N.	2055-2065	10-15	10-15
206	1	N.W.	2065-2075	10-15	10-15
207	1	W.	2075-2085	10-15	10-15
208	1	W.N.W.	2085-2095	10-15	10-15
209	1	N.W.	2095-2105	10-15	10-15
210	1	N.	2105-2115	10-15	10-15
211	1	N.E.	2115-2125	10-15	10-15

INVISIBILITY

1

BASCAT LASER SCATTERING MODULE

 * MONTE CARLO MULTIPLE SCATTERING
 * AEROSOL SCATTERING
 *

PARAMETERS FOR THIS RUN

WAVELENTH= 1.060 MICRONS ALBEDO= .999
 AEROSOL EXTINCTION COEFFICIENT= .4000+001 KM**-1

ELLIPSOIDAL AEROSOL CLOUD
 COORDINATE ORIGIN AT CENTER OF CLOUD
 Z-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH

SOURCE XYZ COORDINATES(KM)= -2000 0000 0000 - .0980
 SOURCE AXIS POLAR ANGLE= 90.000 DEGREES
 SOURCE AXIS AZIMUTH ANGLE= 50.000 DEGREES
 SOURCE APERTURE AND ISOC(MH)= 50.900
 SOURCE BEAM SPREAD ANGLE = .293-004 RADIANS

DETECTOR PARAMETERS

CONE OF VIEW HALF-ANGLE = 1.000 DEGREES
 DETECTOR APERTURE RADIUS = 1.000 CM
 DETECTOR XYZ COORDINATES(KM)= -1200 0000 0000 -.0980
 DETECTOR POLAR ANGLE = 90.000 DEGREES
 DETECTOR AXIS AZIMUTH ANGLE= 180.000 DEGREES

ISOTROPIC REFLECTION FROM GROUND PLANE
 GROUND PLANE 2-COORDINATE ZG(KM)= -100
 GROUND PLANE ALBEDO = .500

CLOUD PARAMETERS
 ELLIPSOID PRINCIPAL XYZ HALF-AXES(KM)
 EULER ANGLES THE PHE, PSE OF ELLIPSOID XYZ AXES = 1000 2000 1000
 OPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES = .8000 1.6000 .8000
 TOTAL .2797-001

STEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE POWER

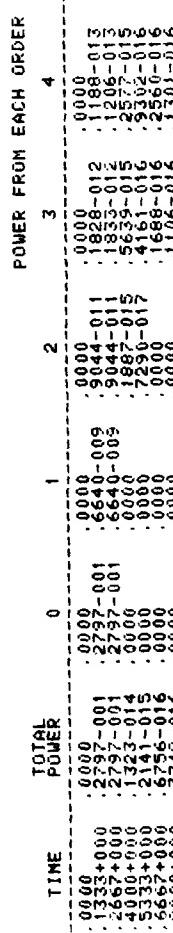
ORDER	STEADY STATE POWER	NUMBER OF PHOTONS
0	.2797-001	.500000+004
1	.66398-009	.500000+004
2	.90441-011	.499999+004
3	.18337-012	.500000+004
4	.12166-013	.500000+004
5	.58910-015	.500000+004
TOTAL	.2797-001	.500000+004

POWER INTO DETECTOR FOR 1 PULSE(S) OF DIFFERENT LENGTH

PULSE NUMBER 1 HAS LENGTH 2667+000 MICROSECONDS

DETECTOR RESPONSE CUTOFF TIME FOR PULSE NUMBER 1 IS .7067+001 MICROSECONDS

DETECTOR RESPONSE, POWER AS A FUNCTION OF TIME, FOR UNIT PULSE POWER



— १८ —

**** EUSSEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

RUN NUMBER 3

	INDIVIDUAL MODULES NNW	SELECTED	
	BEGINNING		ENDING
WAVENUMBER(CM**-1)	1.167		1.167
WAVELENGTH(MICRONS)	8571.428		8571.428
FREQUENCY(GHZ)	35.000		35.000

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

FOG, HAZE AND MIST WITH VIS LT 1 KM. LT 3 KM.
FOG, HAZE AND MIST WITH VIS LT 3 KM.
FOG, HAZE AND MIST LT 3 KM.
FOG, HAZE WITH VIS GE 3 KM.
DUST WITH RAIN AND TSTMS WITH VIS LT 1 KM.
DUST WITH RAIN AND TSTMS WITH VIS GE 3 KM.
DRIZZLE, RAINS WITH VIS LT 1 KM.
DRIZZLE, RAINS WITH VIS GE 3 KM.
SNOW WITH VIS LT 1 KM.
SNOW WITH VIS GE 3 KM.
NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
VIS LT 3 KM AND CEILING HEIGHT LT 300 M.
VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
NO CEILING HEIGHT LT 1000 M.
ALL CONDITIONS COMBINED.

EUROPEAN CLIMATOLOGY FOR LOWLANDS

EUROPEAN CLIMATOLOGY FOR EUROPEAN LOWLANDS									
CLASS NO.	FREQUENCY CLASS (%)	MEAN TEMP (°C)	MEAN RH (%)	MEAN PRESS (KPa)	MEAN V.WS (Km)	MEAN WND SPD (MPS)	MEAN WND DIR (°)	MEAN STD DEV (KM)	MEAN CLD VR (%)
1	1.5-1.9	3.4	71	1010	1.1	1.5	144	1.9	11.5
2	2.0-2.4	4.0	70	1010	1.1	1.5	144	1.9	11.5
3	2.5-2.9	4.6	70	1010	1.1	1.5	144	1.9	11.5
4	3.0-3.4	5.2	70	1010	1.1	1.5	144	1.9	11.5
5	3.5-3.9	5.8	70	1010	1.1	1.5	144	1.9	11.5
6	4.0-4.4	6.4	70	1010	1.1	1.5	144	1.9	11.5
7	4.5-4.9	7.0	70	1010	1.1	1.5	144	1.9	11.5
8	5.0-5.4	7.6	70	1010	1.1	1.5	144	1.9	11.5
9	5.5-5.9	8.2	70	1010	1.1	1.5	144	1.9	11.5
10	6.0-6.4	8.8	70	1010	1.1	1.5	144	1.9	11.5
11	6.5-6.9	9.4	70	1010	1.1	1.5	144	1.9	11.5
12	7.0-7.4	10.0	70	1010	1.1	1.5	144	1.9	11.5
13	7.5-7.9	10.6	70	1010	1.1	1.5	144	1.9	11.5
14	8.0-8.4	11.2	70	1010	1.1	1.5	144	1.9	11.5
15	8.5-8.9	11.8	70	1010	1.1	1.5	144	1.9	11.5
16	9.0-9.4	12.4	70	1010	1.1	1.5	144	1.9	11.5
17	9.5-9.9	13.0	70	1010	1.1	1.5	144	1.9	11.5
18	10.0-10.4	13.6	70	1010	1.1	1.5	144	1.9	11.5
19	10.5-10.9	14.2	70	1010	1.1	1.5	144	1.9	11.5
20	11.0-11.4	14.8	70	1010	1.1	1.5	144	1.9	11.5
21	11.5-11.9	15.4	70	1010	1.1	1.5	144	1.9	11.5
22	12.0-12.4	16.0	70	1010	1.1	1.5	144	1.9	11.5
23	12.5-12.9	16.6	70	1010	1.1	1.5	144	1.9	11.5
24	13.0-13.4	17.2	70	1010	1.1	1.5	144	1.9	11.5
25	13.5-13.9	17.8	70	1010	1.1	1.5	144	1.9	11.5
26	14.0-14.4	18.4	70	1010	1.1	1.5	144	1.9	11.5
27	14.5-14.9	19.0	70	1010	1.1	1.5	144	1.9	11.5
28	15.0-15.4	19.6	70	1010	1.1	1.5	144	1.9	11.5
29	15.5-15.9	20.2	70	1010	1.1	1.5	144	1.9	11.5
30	16.0-16.4	20.8	70	1010	1.1	1.5	144	1.9	11.5
31	16.5-16.9	21.4	70	1010	1.1	1.5	144	1.9	11.5
32	17.0-17.4	22.0	70	1010	1.1	1.5	144	1.9	11.5
33	17.5-17.9	22.6	70	1010	1.1	1.5	144	1.9	11.5
34	18.0-18.4	23.2	70	1010	1.1	1.5	144	1.9	11.5
35	18.5-18.9	23.8	70	1010	1.1	1.5	144	1.9	11.5
36	19.0-19.4	24.4	70	1010	1.1	1.5	144	1.9	11.5
37	19.5-19.9	25.0	70	1010	1.1	1.5	144	1.9	11.5
38	20.0-20.4	25.6	70	1010	1.1	1.5	144	1.9	11.5
39	20.5-20.9	26.2	70	1010	1.1	1.5	144	1.9	11.5
40	21.0-21.4	26.8	70	1010	1.1	1.5	144	1.9	11.5
41	21.5-21.9	27.4	70	1010	1.1	1.5	144	1.9	11.5
42	22.0-22.4	28.0	70	1010	1.1	1.5	144	1.9	11.5
43	22.5-22.9	28.6	70	1010	1.1	1.5	144	1.9	11.5
44	23.0-23.4	29.2	70	1010	1.1	1.5	144	1.9	11.5
45	23.5-23.9	29.8	70	1010	1.1	1.5	144	1.9	11.5
46	24.0-24.4	30.4	70	1010	1.1	1.5	144	1.9	11.5
47	24.5-24.9	31.0	70	1010	1.1	1.5	144	1.9	11.5
48	25.0-25.4	31.6	70	1010	1.1	1.5	144	1.9	11.5
49	25.5-25.9	32.2	70	1010	1.1	1.5	144	1.9	11.5
50	26.0-26.4	32.8	70	1010	1.1	1.5	144	1.9	11.5
51	26.5-26.9	33.4	70	1010	1.1	1.5	144	1.9	11.5
52	27.0-27.4	34.0	70	1010	1.1	1.5	144	1.9	11.5
53	27.5-27.9	34.6	70	1010	1.1	1.5	144	1.9	11.5
54	28.0-28.4	35.2	70	1010	1.1	1.5	144	1.9	11.5
55	28.5-28.9	35.8	70	1010	1.1	1.5	144	1.9	11.5
56	29.0-29.4	36.4	70	1010	1.1	1.5	144	1.9	11.5
57	29.5-29.9	37.0	70	1010	1.1	1.5	144	1.9	11.5
58	30.0-30.4	37.6	70	1010	1.1	1.5	144	1.9	11.5
59	30.5-30.9	38.2	70	1010	1.1	1.5	144	1.9	11.5
60	31.0-31.4	38.8	70	1010	1.1	1.5	144	1.9	11.5
61	31.5-31.9	39.4	70	1010	1.1	1.5	144	1.9	11.5
62	32.0-32.4	40.0	70	1010	1.1	1.5	144	1.9	11.5
63	32.5-32.9	40.6	70	1010	1.1	1.5	144	1.9	11.5
64	33.0-33.4	41.2	70	1010	1.1	1.5	144	1.9	11.5
65	33.5-33.9	41.8	70	1010	1.1	1.5	144	1.9	11.5
66	34.0-34.4	42.4	70	1010	1.1	1.5	144	1.9	11.5
67	34.5-34.9	43.0	70	1010	1.1	1.5	144	1.9	11.5
68	35.0-35.4	43.6	70	1010	1.1	1.5	144	1.9	11.5
69	35.5-35.9	44.2	70	1010	1.1	1.5	144	1.9	11.5
70	36.0-36.4	44.8	70	1010	1.1	1.5	144	1.9	11.5
71	36.5-36.9	45.4	70	1010	1.1	1.5	144	1.9	11.5
72	37.0-37.4	46.0	70	1010	1.1	1.5	144	1.9	11.5
73	37.5-37.9	46.6	70	1010	1.1	1.5	144	1.9	11.5
74	38.0-38.4	47.2	70	1010	1.1	1.5	144	1.9	11.5
75	38.5-38.9	47.8	70	1010	1.1	1.5	144	1.9	11.5
76	39.0-39.4	48.4	70	1010	1.1	1.5	144	1.9	11.5
77	39.5-39.9	49.0	70	1010	1.1	1.5	144	1.9	11.5
78	40.0-40.4	49.6	70	1010	1.1	1.5	144	1.9	11.5
79	40.5-40.9	50.2	70	1010	1.1	1.5	144	1.9	11.5
80	41.0-41.4	50.8	70	1010	1.1	1.5	144	1.9	11.5
81	41.5-41.9	51.4	70	1010	1.1	1.5	144	1.9	11.5
82	42.0-42.4	52.0	70	1010	1.1	1.5	144	1.9	11.5
83	42.5-42.9	52.6	70	1010	1.1	1.5	144	1.9	11.5
84	43.0-43.4	53.2	70	1010	1.1	1.5	144	1.9	11.5
85	43.5-43.9	53.8	70	1010	1.1	1.5	144	1.9	11.5
86	44.0-44.4	54.4	70	1010	1.1	1.5	144	1.9	11.5
87	44.5-44.9	55.0	70	1010	1.1	1.5	144	1.9	11.5
88	45.0-45.4	55.6	70	1010	1.1	1.5	144	1.9	11.5
89	45.5-45.9	56.2	70	1010	1.1	1.5	144	1.9	11.5
90	46.0-46.4	56.8	70	1010	1.1	1.5	144	1.9	11.5
91	46.5-46.9	57.4	70	1010	1.1	1.5	144	1.9	11.5
92	47.0-47.4	58.0	70	1010	1.1	1.5	144	1.9	11.5
93	47.5-47.9	58.6	70	1010	1.1	1.5	144	1.9	11.5
94	48.0-48.4	59.2	70	1010	1.1	1.5	144	1.9	11.5
95	48.5-48.9	59.8	70	1010	1.1	1.5	144	1.9	11.5
96	49.0-49.4	60.4	70	1010	1.1	1.5	144	1.9	11.5
97	49.5-49.9	61.0	70	1010	1.1	1.5	144	1.9	11.5
98	50.0-50.4	61.6	70	1010	1.1	1.5	144	1.9	11.5
99	50.5-50.9	62.2	70	1010	1.1	1.5	144	1.9	11.5
100	51.0-51.4	62.8	70	1010	1.1	1.5	144	1.9	11.5
101	51.5-51.9	63.4	70	1010	1.1	1.5	144	1.9	11.5
102	52.0-52.4	64.0	70	1010	1.1	1.5	144	1.9	11.5
103	52.5-52.9	64.6	70	1010	1.1	1.5	144	1.9	11.5
104	53.0-53.4	65.2	70	1010	1.1	1.5	144	1.9	11.5
105	53.5-53.9	65.8	70	1010	1.1	1.5	144	1.9	11.5
106	54.0-54.4	66.4	70	1010	1.1	1.5	144	1.9	11.5
107	54.5-54.9	67.0	70	1010	1.1	1.5	144	1.9	11.5
108	55.0-55.4	67.6	70	1010	1.1	1.5	144	1.9	11.5
109	55.5-55.9	68.2	70	1010	1.1	1.5	144	1.9	11.5
110	56.0-56.4	68.8	70	1010	1.1	1.5	144	1.9	11.5
111	56.5-56.9	69.4	70	1010	1.1	1.5	144	1.9	11.5
112	57.0-57.4	70.0	70	1010	1.1	1.5	144	1.9	11.5
113	57.5-57.9	70.6	70	1010	1.1	1.5	144	1.9	11.5
114	58.0-58.4	71.2	70	1010	1.1	1.5	144	1.9	11.5
115	58.5-58.9	71.8	70	1010	1.1	1.5	144	1.9	11.5
116	59.0-59.4	72.4	70	1010	1.1	1.5	144	1.9	11.5
117	59.5-59.9	73.0	70	1010	1.1	1.5	144	1.9	11.5
118	60.0-60.4	73.6	70	1010	1.1	1.5	144	1.9	11.5
119	60.5-60.9	74.2	70	1010	1.1	1.5	144	1.9	11.5
120	61.0-61.4	74.8	70	1010	1.1	1.5	144	1.9	11.5
121	61.5-61.9	75.4	70	1010	1.1	1.5	144	1.9	11.5
122	62.0-62.4	76.0	70	1010	1.1	1.5	144	1.9	11.5
123	62.5-62.9	76.6	70	1010	1.1	1.5	144	1.9	11.5
124	63.0-63.4	77.2	70	1010	1.1	1.5	144	1.9	11.5
125	63.5-63.9	77.8	70	1010	1.1	1.5	144	1.9	11.5
126	64.0-64.4	78.4	70	1010	1.1	1.5	144	1.9	11.5
127	64.5-64.9	79.0	70	1010	1.1	1.5	144	1.9	11.5
128	65.0-65.4	79.6	70	1010	1.1	1.5	144	1.9	11.5
129	65.5-65.9	80.2	70	1010	1.1	1.5	144	1.9	11.5
130	66.0-66.4	80.8	70	1010	1.1	1.5	144	1	

DURING SPRING AT 03-09 (LST).

- 130 -

NEAR MILLIMETER WAVE MODULE

	5.300 DEGREES C
TEMPERATURE	1.015 .000
PRESSURE	.6 .300
ABSOLUTE HUMIDITY	G/H**3
FOG DENSITY	.500 G/H**3
RAIN RATE	.5 .000 MM/HR
SNOW RATE	.000 MM/HR
FREQUENCY	.35 .000 GHZ
PATH LENGTH	.400 KM
GAS ABSORPTION	6838-001 DB/KM
FOG EXTINCTION	.4565+000 DB/KM
RAIN EXTINCTION	.1252+001 DB/KM
SNOW EXTINCTION	.0010 DB/KM
TRANSMISSION	.8413+000
FOG BACKSCATTER	.5724-010 N**2/H**3
RAIN BACKSCATTER	.1226-003 N**2/H**3
SNOW BACKSCATTER	.0000 N**2/H**3
TOTAL BACKSCATTER	.1026-003 N**2/H**3

	5.300 DEGREES C
TEMPERATURE	1.015 .000
PRESSURE	.6 .300
ABSOLUTE HUMIDITY	G/H**3
FOG DENSITY	.500 G/H**3
RAIN RATE	.5 .000 MM/HR
SNOW RATE	.000 MM/HR
FREQUENCY	.35 .000 GHZ
PATH LENGTH	.400 KM
GAS ABSORPTION	6838-001 DB/KM
FOG EXTINCTION	.4565+000 DB/KM
RAIN EXTINCTION	.0000 DB/KM
SNOW EXTINCTION	.4508+001 DB/KM
TRANSMISSION	.6175+000
FOG BACKSCATTER	.5724-010 N**2/H**3
RAIN BACKSCATTER	.0909-010 N**2/H**3
SNOW BACKSCATTER	.3227-003 N**2/H**3
TOTAL BACKSCATTER	.3227-003 N**2/H**3

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .5195+000

RUN NUMBER 4

**** EOSAEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED	SPOT	LT4M	BEGINNING	ENDING
WAVENUMBER(CH**-1)			2010.000	2710.000
WAVELENGTH(MICRONS)			3.690	4.975
FREQUENCY(GHZ)			60300.000	81300.000

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

EUROPEAN CLIMATOLOGY FOR CLIMATE LOWLANDS

DURING SPRING AT 03-09 (LST).

VISIBLE KM
9.30 KM

SPOT CONTRAST MODULE

SPOT CONTROL CARDS READ FOR THIS RUN:

	INPUT	FUNCTION
WAVELENGTH	350000	7800+000
	360000	8300+000
	370000	8700+000
	380000	9200+000
	390000	9600+000
	400000	9900+000
	410000	9700+000
	420000	9300+000
	430000	9600+000
	440000	9500+000
	450000	9400+000
	460000	9300+000
	470000	9300+000
	480000	9300+000
	490000	9300+000
	500000	9300+000

***** DIVISION LIMITS CHANGED FROM RESPONSE FUNCTION *****

SPOT DIAGNOSTIC MESSAGES FOLLOW:

1. NO DIRECT SUNLIGHT INCIDENT WITHIN RECEIVER'S FIELD OF VIEW.

***** DEFINITION OF CONTROL PARAMETERS FOLLOWS: *****

PARAMETER	VALUE	DESCRIPTION
ISORC	3	SUNLIGHT AND EMISSION
ITARG	2	TARGET REFLECTANCE / EMISSION
MODEL	4	SUBARCTIC SUMMER
IHAZE	2	
CLOUD BOTTOM HEIGHT	.000 KM	

WAVELENGTH (MICRONS)	WAVENUMBER (CM ⁻¹)	SOURCE INTENSITIES		TARGET SOURCE STRENGTH (WATTS M ⁻² MICRON ⁻¹ SR ⁻¹)	GROUND SOURCE STRENGTH (WATTS M ⁻² MICRON ⁻¹ SR ⁻¹)
		SUNLIGHT SOURCE (WATTS M ⁻² MICRON ⁻¹)	SOURCE (WATTS M ⁻² MICRON ⁻¹ SR ⁻¹)		
2.010	2273	0.000	0.000	0.000	0.000
2.073	2200	0.000	0.000	0.000	0.000
2.130	2150	0.000	0.000	0.000	0.000
2.188	2125	0.000	0.000	0.000	0.000
2.246	2100	0.000	0.000	0.000	0.000
2.293	2075	0.000	0.000	0.000	0.000
2.341	2050	0.000	0.000	0.000	0.000
2.389	2025	0.000	0.000	0.000	0.000
2.437	2000	0.000	0.000	0.000	0.000
2.484	1975	0.000	0.000	0.000	0.000
2.532	1950	0.000	0.000	0.000	0.000
2.579	1925	0.000	0.000	0.000	0.000
2.627	1900	0.000	0.000	0.000	0.000
2.674	1875	0.000	0.000	0.000	0.000
2.721	1850	0.000	0.000	0.000	0.000
2.769	1825	0.000	0.000	0.000	0.000
2.816	1800	0.000	0.000	0.000	0.000
2.863	1775	0.000	0.000	0.000	0.000
2.911	1750	0.000	0.000	0.000	0.000
2.958	1725	0.000	0.000	0.000	0.000
3.006	1700	0.000	0.000	0.000	0.000
3.053	1675	0.000	0.000	0.000	0.000
3.101	1650	0.000	0.000	0.000	0.000
3.148	1625	0.000	0.000	0.000	0.000
3.196	1600	0.000	0.000	0.000	0.000
3.243	1575	0.000	0.000	0.000	0.000
3.291	1550	0.000	0.000	0.000	0.000
3.338	1525	0.000	0.000	0.000	0.000
3.386	1500	0.000	0.000	0.000	0.000
3.433	1475	0.000	0.000	0.000	0.000
3.481	1450	0.000	0.000	0.000	0.000
3.528	1425	0.000	0.000	0.000	0.000
3.576	1400	0.000	0.000	0.000	0.000
3.623	1375	0.000	0.000	0.000	0.000
3.671	1350	0.000	0.000	0.000	0.000
3.718	1325	0.000	0.000	0.000	0.000
3.766	1300	0.000	0.000	0.000	0.000
3.813	1275	0.000	0.000	0.000	0.000
3.861	1250	0.000	0.000	0.000	0.000
3.908	1225	0.000	0.000	0.000	0.000
3.956	1200	0.000	0.000	0.000	0.000
4.003	1175	0.000	0.000	0.000	0.000
4.051	1150	0.000	0.000	0.000	0.000
4.098	1125	0.000	0.000	0.000	0.000
4.146	1100	0.000	0.000	0.000	0.000
4.193	1075	0.000	0.000	0.000	0.000
4.241	1050	0.000	0.000	0.000	0.000
4.288	1025	0.000	0.000	0.000	0.000
4.336	1000	0.000	0.000	0.000	0.000
4.383	975	0.000	0.000	0.000	0.000
4.431	950	0.000	0.000	0.000	0.000
4.478	925	0.000	0.000	0.000	0.000
4.526	900	0.000	0.000	0.000	0.000
4.573	875	0.000	0.000	0.000	0.000
4.621	850	0.000	0.000	0.000	0.000
4.668	825	0.000	0.000	0.000	0.000
4.716	800	0.000	0.000	0.000	0.000
4.763	775	0.000	0.000	0.000	0.000
4.811	750	0.000	0.000	0.000	0.000
4.858	725	0.000	0.000	0.000	0.000
4.906	700	0.000	0.000	0.000	0.000
4.953	675	0.000	0.000	0.000	0.000
5.001	650	0.000	0.000	0.000	0.000
5.048	625	0.000	0.000	0.000	0.000
5.096	600	0.000	0.000	0.000	0.000
5.143	575	0.000	0.000	0.000	0.000
5.191	550	0.000	0.000	0.000	0.000
5.238	525	0.000	0.000	0.000	0.000
5.286	500	0.000	0.000	0.000	0.000
5.333	475	0.000	0.000	0.000	0.000
5.381	450	0.000	0.000	0.000	0.000
5.428	425	0.000	0.000	0.000	0.000
5.476	400	0.000	0.000	0.000	0.000
5.523	375	0.000	0.000	0.000	0.000
5.571	350	0.000	0.000	0.000	0.000
5.618	325	0.000	0.000	0.000	0.000
5.666	300	0.000	0.000	0.000	0.000
5.713	275	0.000	0.000	0.000	0.000
5.761	250	0.000	0.000	0.000	0.000
5.808	225	0.000	0.000	0.000	0.000
5.856	200	0.000	0.000	0.000	0.000
5.903	175	0.000	0.000	0.000	0.000
5.951	150	0.000	0.000	0.000	0.000
6.000	125	0.000	0.000	0.000	0.000
6.047	100	0.000	0.000	0.000	0.000
6.095	75	0.000	0.000	0.000	0.000
6.143	50	0.000	0.000	0.000	0.000
6.190	25	0.000	0.000	0.000	0.000
6.238	0	0.000	0.000	0.000	0.000

COMPONENTS FOR RADIANCE FROM TARGET					
(WATTS M-2 MICRON-1 SR-1)					
WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET EMISSION	TARGET REFLECTANCE	PARTIAL ATMOSPHERIC EMISSION	PARTIAL PATH RADIANCE
4.9751+0.00	2.010	8.3677-0.01	3.5950-0.03	6.9558-0.01	2.6453-0.04
4.8309+0.00	2.070	8.0848-0.01	3.4220-0.03	6.1338-0.01	1.0494-0.04
4.6948+0.00	2.130	8.8964-0.01	3.6302-0.03	6.1233-0.01	1.1211+0.00
4.5662+0.00	2.190	6.8910-0.01	3.1827-0.03	4.7137-0.01	4.8794-0.04
4.4444+0.00	2.250	6.8915-0.01	3.0515-0.03	4.8835-0.01	4.8835-0.04
4.3221+0.00	2.310	6.8914-0.01	2.9126-0.03	4.6436-0.01	9.6050-0.01
4.2044+0.00	2.370	6.8914-0.01	2.7676-0.03	4.1546-0.01	9.4575-0.01
4.0912+0.00	2.430	6.8914-0.01	2.6246-0.03	3.6246-0.01	9.0877-0.01
3.9871+0.00	2.490	6.8914-0.01	2.4816-0.03	3.0346-0.01	8.1542-0.01
3.8821+0.00	2.550	6.8914-0.01	2.3386-0.03	2.4386-0.01	6.1183-0.01
3.7763+0.00	2.610	6.8914-0.01	2.1956-0.03	1.8486-0.01	4.4356-0.01
3.6700+0.00	2.670	6.8914-0.01	2.0526-0.03	1.2526-0.01	3.0444-0.01
3.5630+0.00	2.730	6.8914-0.01	1.9096-0.03	6.9096-0.01	2.4474-0.01
3.4554+0.00	2.790	6.8914-0.01	1.7666-0.03	1.5666-0.01	1.2454-0.01
3.3473+0.00	2.850	6.8914-0.01	1.6236-0.03	1.4236-0.01	1.0443-0.01
3.2390+0.00	2.910	6.8914-0.01	1.4806-0.03	1.2806-0.01	8.4440-0.01
3.1307+0.00	2.970	6.8914-0.01	1.3376-0.03	1.1376-0.01	7.2456-0.01
3.0221+0.00	3.030	6.8914-0.01	1.1946-0.03	9.9946-0.01	3.2253-0.01
2.9134+0.00	3.090	6.8914-0.01	1.0516-0.03	9.9916-0.01	3.0444-0.01
2.8046+0.00	3.150	6.8914-0.01	9.0886-0.03	9.9986-0.01	2.8646-0.01
2.6958+0.00	3.210	6.8914-0.01	7.6256-0.03	9.9958-0.01	2.6858-0.01
2.5869+0.00	3.270	6.8914-0.01	6.1626-0.03	9.9969-0.01	2.5069-0.01
2.4779+0.00	3.330	6.8914-0.01	4.7006-0.03	9.9979-0.01	2.3279-0.01
2.3688+0.00	3.390	6.8914-0.01	3.2376-0.03	9.9988-0.01	2.1488-0.01
2.2596+0.00	3.450	6.8914-0.01	1.7756-0.03	9.9996-0.01	1.9696-0.01
2.1504+0.00	3.510	6.8914-0.01	1.3126-0.03	9.9904-0.01	1.7804-0.01
2.0411+0.00	3.570	6.8914-0.01	8.6596-0.03	9.9911-0.01	1.5911-0.01
1.9318+0.00	3.630	6.8914-0.01	4.2066-0.03	9.9921-0.01	1.4121-0.01
1.8225+0.00	3.690	6.8914-0.01	-0.1436-0.03	9.9931-0.01	1.2325-0.01
1.7132+0.00	3.750	6.8914-0.01	-4.6866-0.03	9.9941-0.01	1.0532-0.01
1.6039+0.00	3.810	6.8914-0.01	-9.3296-0.03	9.9951-0.01	8.7639-0.01
1.4946+0.00	3.870	6.8914-0.01	-14.9726-0.03	9.9961-0.01	7.4846-0.01
1.3853+0.00	3.930	6.8914-0.01	-20.6156-0.03	9.9971-0.01	6.2053-0.01
1.2760+0.00	3.990	6.8914-0.01	-26.2586-0.03	9.9981-0.01	4.9260-0.01
1.1667+0.00	4.050	6.8914-0.01	-31.8916-0.03	9.9991-0.01	3.6467-0.01
1.0574+0.00	4.110	6.8914-0.01	-37.5246-0.03	9.9901-0.01	2.3674-0.01
9.4881+0.00	4.170	6.8914-0.01	-43.1576-0.03	9.9911-0.01	1.0881-0.01
8.4288+0.00	4.230	6.8914-0.01	-48.7906-0.03	9.9921-0.01	8.5088-0.01
7.3695+0.00	4.290	6.8914-0.01	-54.4236-0.03	9.9931-0.01	6.2295-0.01
6.3102+0.00	4.350	6.8914-0.01	-59.0566-0.03	9.9941-0.01	3.9410-0.01
5.2509+0.00	4.410	6.8914-0.01	-64.6896-0.03	9.9951-0.01	2.6609-0.01
4.1916+0.00	4.470	6.8914-0.01	-69.3226-0.03	9.9961-0.01	1.3716-0.01
3.1323+0.00	4.530	6.8914-0.01	-74.9556-0.03	9.9971-0.01	0.0823-0.01
2.0730+0.00	4.590	6.8914-0.01	-79.5886-0.03	9.9981-0.01	-0.8130-0.01
1.0137+0.00	4.650	6.8914-0.01	-84.2216-0.03	9.9991-0.01	-1.6337-0.01
0.9544+0.00	4.710	6.8914-0.01	-88.8546-0.03	9.9901-0.01	-2.4544-0.01
0.8951+0.00	4.770	6.8914-0.01	-93.4876-0.03	9.9911-0.01	-3.2751-0.01
0.8358+0.00	4.830	6.8914-0.01	-98.1206-0.03	9.9921-0.01	-4.0958-0.01
0.7765+0.00	4.890	6.8914-0.01	-102.7536-0.03	9.9931-0.01	-4.9165-0.01
0.7172+0.00	4.950	6.8914-0.01	-107.3866-0.03	9.9941-0.01	-5.7372-0.01
0.6579+0.00	5.010	6.8914-0.01	-111.0196-0.03	9.9951-0.01	-6.5579-0.01
0.5986+0.00	5.070	6.8914-0.01	-115.6526-0.03	9.9961-0.01	-7.3786-0.01
0.5393+0.00	5.130	6.8914-0.01	-120.2856-0.03	9.9971-0.01	-8.1993-0.01
0.4799+0.00	5.190	6.8914-0.01	-124.9186-0.03	9.9981-0.01	-8.9199-0.01
0.4206+0.00	5.250	6.8914-0.01	-129.5516-0.03	9.9991-0.01	-9.7306-0.01
0.3613+0.00	5.310	6.8914-0.01	-134.1846-0.03	9.9901-0.01	-10.5413-0.01
0.3020+0.00	5.370	6.8914-0.01	-138.8176-0.03	9.9911-0.01	-11.3520-0.01
0.2427+0.00	5.430	6.8914-0.01	-143.4506-0.03	9.9921-0.01	-12.1627-0.01
0.1834+0.00	5.490	6.8914-0.01	-148.0836-0.03	9.9931-0.01	-12.9734-0.01
0.1241+0.00	5.550	6.8914-0.01	-152.7166-0.03	9.9941-0.01	-13.7841-0.01
0.0648+0.00	5.610	6.8914-0.01	-157.3496-0.03	9.9951-0.01	-14.5948-0.01
0.0055+0.00	5.670	6.8914-0.01	-161.9826-0.03	9.9961-0.01	-15.4055-0.01
-0.1544-0.01	5.730	6.8914-0.01	-166.6156-0.03	9.9971-0.01	-16.2154-0.01
-0.3151-0.01	5.790	6.8914-0.01	-171.2486-0.03	9.9981-0.01	-17.0251-0.01
-0.4758-0.01	5.850	6.8914-0.01	-175.8816-0.03	9.9991-0.01	-17.8358-0.01
-0.6365-0.01	5.910	6.8914-0.01	-180.5146-0.03	9.9901-0.01	-18.6465-0.01
-0.7972-0.01	5.970	6.8914-0.01	-185.1476-0.03	9.9911-0.01	-19.4572-0.01
-0.9579-0.01	6.030	6.8914-0.01	-189.7806-0.03	9.9921-0.01	-20.2679-0.01
-1.1186-0.01	6.090	6.8914-0.01	-194.4136-0.03	9.9931-0.01	-21.0786-0.01
-1.2793-0.01	6.150	6.8914-0.01	-199.0466-0.03	9.9941-0.01	-21.8893-0.01
-1.4399-0.01	6.210	6.8914-0.01	-203.6796-0.03	9.9951-0.01	-22.6999-0.01
-1.5906-0.01	6.270	6.8914-0.01	-208.3126-0.03	9.9961-0.01	-23.5006-0.01
-1.7513-0.01	6.330	6.8914-0.01	-212.9456-0.03	9.9971-0.01	-24.3113-0.01
-1.9120-0.01	6.390	6.8914-0.01	-217.5786-0.03	9.9981-0.01	-25.1220-0.01
-2.0727-0.01	6.450	6.8914-0.01	-222.2116-0.03	9.9991-0.01	-25.9327-0.01
-2.2334-0.01	6.510	6.8914-0.01	-226.8446-0.03	9.9901-0.01	-26.7434-0.01
-2.3941-0.01	6.570	6.8914-0.01	-231.4776-0.03	9.9911-0.01	-27.5541-0.01
-2.5548-0.01	6.630	6.8914-0.01	-236.1106-0.03	9.9921-0.01	-28.3648-0.01
-2.7155-0.01	6.690	6.8914-0.01	-240.7436-0.03	9.9931-0.01	-29.1755-0.01
-2.8762-0.01	6.750	6.8914-0.01	-245.3766-0.03	9.9941-0.01	-29.9862-0.01
-3.0369-0.01	6.810	6.8914-0.01	-249.0096-0.03	9.9951-0.01	-30.7969-0.01
-3.1976-0.01	6.870	6.8914-0.01	-253.6426-0.03	9.9961-0.01	-31.6076-0.01
-3.3583-0.01	6.930	6.8914-0.01	-258.2756-0.03	9.9971-0.01	-32.4183-0.01
-3.5190-0.01	6.990	6.8914-0.01	-262.9086-0.03	9.9981-0.01	-33.2290-0.01
-3.6797-0.01	7.050	6.8914-0.01	-267.5416-0.03	9.9991-0.01	-34.0397-0.01
-3.8304-0.01	7.110	6.8914-0.01	-272.1746-0.03	9.9901-0.01	-34.8404-0.01
-4.9751+0.00	7.170	6.8914-0.01	-276.8076-0.03	9.9911-0.01	-35.6511-0.01
-4.8309+0.00	7.230	6.8914-0.01	-281.4406-0.03	9.9921-0.01	-36.4619-0.01
-4.6948+0.00	7.290	6.8914-0.01	-286.0736-0.03	9.9931-0.01	-37.2726-0.01
-4.5662+0.00	7.350	6.8914-0.01	-290.7066-0.03	9.9941-0.01	-38.0832-0.01
-4.4444+0.00	7.410	6.8914-0.01	-295.3396-0.03	9.9951-0.01	-38.8944-0.01
-4.3221+0.00	7.470	6.8914-0.01	-300.9726-0.03	9.9961-0.01	-39.7051-0.01
-4.2003+0.00	7.530	6.8914-0.01	-305.6056-0.03	9.9971-0.01	-40.5158-0.01
-4.0780+0.00	7.590	6.8914-0.01	-310.2386-0.03	9.9981-0.01	-41.3265-0.01
-3.9557+0.00	7.650	6.8914-0.01	-314.8716-0.03	9.9991-0.01	-42.1372-0.01
-3.8334+0.00	7.710	6.8914-0.01	-319.5046-0.03	9.9901-0.01	-42.9479-0.01
-3.7111+0.00	7.770	6.8914-0.01	-324.1376-0.03	9.9911-0.01	-43.7586-0.01
-3.5888+0.00	7.830	6.8914-0.01	-328.7706-0.03	9.9921-0.01	-44.5693-0.01
-3.4665+0.00	7.890	6.8914-0.01	-333.4036-0.03	9.9931-0.01	-45.3799-0.01
-3.3442+0.00	7.950	6.8914-0.01	-338.0366-0.03	9.9941-0.01	-46.1896-0.01
-3.2219+0.00	8.010	6.8914-0.01	-342.6696-0.03	9.9951-0.01	-46.9993-0.01
-3.1096+0.00	8.070	6.8914-0.01	-347.3026-0.03	9.9961-0.01	-47.8090-0.01
-2.9973+0.00	8.130	6.8914-0.01	-351.9356-0.03	9.9971-0.01	-48.6187-0.01
-2.8850+0.00	8.190	6.8914-0.01	-356.5686-0.03	9.9981-0.01	-49.4284-0.01
-2.7727+0.00	8.250	6.8914-0.01	-361.2016-0.03	9.9991-0.01	-50.2381-0.01
-2.6604+0.00	8.310	6.8914-0.01	-365.8346-0.03	9.9901-0.01	-51.0478-0.01
-2.5481+0.00	8.370	6.8914-0.01	-370.4676-0.03	9.9911-0.01	-51.8575-0.01
-2.4358+0.00	8.430	6.8914-0.01	-375.1006-0.03	9.9921-0.01	-52.6672-0.01
-2.3235+0.00	8.490	6.8914-0.01	-379.7336-0.03	9.9931-0.01	-53.4769-0.01
-2.2112+0.00	8.550	6.8914-0.01	-384.3666-0.03	9.9941-0.01	-54.2866-0.01
-2.0989+0.00	8.610	6.8914-0.01	-388.9996-0.03</		

DIRECT SUNLIGHT		SUNLIGHT FLUX
(WATTS M ⁻² MICRON ⁻¹)		SUNLIGHT SOURCE STRENGTH
WAVELENGTH (MICRONS)	WAVENUMBER (CM ⁻¹)	
9.751+000	3.8787+000	.0000
9.750+000	3.87846+000	.0000
9.749+000	3.87845+000	.0000
9.748+000	3.87741+000	.0000
9.747+000	3.87742+000	.0000
9.746+000	3.877482+000	.0000
9.745+000	3.877488+000	.0000
9.744+000	3.87890+000	.0000
9.743+000	3.87916+000	.0000
9.742+000	3.87960+001	.0000
9.741+000	3.87996+001	.0000
9.740+000	3.88032+001	.0000
9.739+000	3.88067+001	.0000
9.738+000	3.88102+001	.0000
9.737+000	3.88137+001	.0000
9.736+000	3.88172+001	.0000
9.735+000	3.88207+001	.0000
9.734+000	3.88242+001	.0000
9.733+000	3.88277+001	.0000
9.732+000	3.88312+001	.0000
9.731+000	3.88347+001	.0000
9.730+000	3.88382+001	.0000
9.729+000	3.88417+001	.0000
9.728+000	3.88452+001	.0000
9.727+000	3.88487+001	.0000
9.726+000	3.88522+001	.0000
9.725+000	3.88557+001	.0000
9.724+000	3.88592+001	.0000
9.723+000	3.88627+001	.0000
9.722+000	3.88662+001	.0000
9.721+000	3.88697+001	.0000
9.720+000	3.88732+001	.0000
9.719+000	3.88767+001	.0000
9.718+000	3.88802+001	.0000
9.717+000	3.88837+001	.0000
9.716+000	3.88872+001	.0000
9.715+000	3.88907+001	.0000
9.714+000	3.88942+001	.0000
9.713+000	3.88977+001	.0000
9.712+000	3.89012+001	.0000
9.711+000	3.89047+001	.0000
9.710+000	3.89082+001	.0000
9.709+000	3.89117+001	.0000
9.708+000	3.89152+001	.0000
9.707+000	3.89187+001	.0000
9.706+000	3.89222+001	.0000
9.705+000	3.89257+001	.0000
9.704+000	3.89292+001	.0000
9.703+000	3.89327+001	.0000
9.702+000	3.89362+001	.0000
9.701+000	3.89397+001	.0000
9.700+000	3.89432+001	.0000
9.699+000	3.89467+001	.0000
9.698+000	3.89502+001	.0000
9.697+000	3.89537+001	.0000
9.696+000	3.89572+001	.0000
9.695+000	3.89607+001	.0000
9.694+000	3.89642+001	.0000
9.693+000	3.89677+001	.0000
9.692+000	3.89712+001	.0000
9.691+000	3.89747+001	.0000
9.690+000	3.89782+001	.0000
9.689+000	3.89817+001	.0000
9.688+000	3.89852+001	.0000
9.687+000	3.89887+001	.0000
9.686+000	3.89922+001	.0000
9.685+000	3.89957+001	.0000
9.684+000	3.89992+001	.0000
9.683+000	3.90027+001	.0000
9.682+000	3.90062+001	.0000
9.681+000	3.90097+001	.0000
9.680+000	3.90132+001	.0000
9.679+000	3.90167+001	.0000
9.678+000	3.90202+001	.0000
9.677+000	3.90237+001	.0000
9.676+000	3.90272+001	.0000
9.675+000	3.90307+001	.0000
9.674+000	3.90342+001	.0000
9.673+000	3.90377+001	.0000
9.672+000	3.90412+001	.0000
9.671+000	3.90447+001	.0000
9.670+000	3.90482+001	.0000
9.669+000	3.90517+001	.0000
9.668+000	3.90552+001	.0000
9.667+000	3.90587+001	.0000
9.666+000	3.90622+001	.0000
9.665+000	3.90657+001	.0000
9.664+000	3.90692+001	.0000
9.663+000	3.90727+001	.0000
9.662+000	3.90762+001	.0000
9.661+000	3.90797+001	.0000
9.660+000	3.90832+001	.0000
9.659+000	3.90867+001	.0000
9.658+000	3.90902+001	.0000
9.657+000	3.90937+001	.0000
9.656+000	3.90972+001	.0000
9.655+000	3.91007+001	.0000
9.654+000	3.91042+001	.0000
9.653+000	3.91077+001	.0000
9.652+000	3.91112+001	.0000
9.651+000	3.91147+001	.0000
9.650+000	3.91182+001	.0000
9.649+000	3.91217+001	.0000
9.648+000	3.91252+001	.0000
9.647+000	3.91287+001	.0000
9.646+000	3.91322+001	.0000
9.645+000	3.91357+001	.0000
9.644+000	3.91392+001	.0000
9.643+000	3.91427+001	.0000
9.642+000	3.91462+001	.0000
9.641+000	3.91497+001	.0000
9.640+000	3.91532+001	.0000
9.639+000	3.91567+001	.0000
9.638+000	3.91602+001	.0000
9.637+000	3.91637+001	.0000
9.636+000	3.91672+001	.0000
9.635+000	3.91707+001	.0000
9.634+000	3.91742+001	.0000
9.633+000	3.91777+001	.0000
9.632+000	3.91812+001	.0000
9.631+000	3.91847+001	.0000
9.630+000	3.91882+001	.0000
9.629+000	3.91917+001	.0000
9.628+000	3.91952+001	.0000
9.627+000	3.91987+001	.0000
9.626+000	3.92022+001	.0000
9.625+000	3.92057+001	.0000
9.624+000	3.92092+001	.0000
9.623+000	3.92127+001	.0000
9.622+000	3.92162+001	.0000
9.621+000	3.92197+001	.0000
9.620+000	3.92232+001	.0000
9.619+000	3.92267+001	.0000
9.618+000	3.92302+001	.0000
9.617+000	3.92337+001	.0000
9.616+000	3.92372+001	.0000
9.615+000	3.92407+001	.0000
9.614+000	3.92442+001	.0000
9.613+000	3.92477+001	.0000
9.612+000	3.92512+001	.0000
9.611+000	3.92547+001	.0000
9.610+000	3.92582+001	.0000
9.609+000	3.92617+001	.0000
9.608+000	3.92652+001	.0000
9.607+000	3.92687+001	.0000
9.606+000	3.92722+001	.0000
9.605+000	3.92757+001	.0000
9.604+000	3.92792+001	.0000
9.603+000	3.92827+001	.0000
9.602+000	3.92862+001	.0000
9.601+000	3.92897+001	.0000
9.600+000	3.92932+001	.0000
9.599+000	3.92967+001	.0000
9.598+000	3.93002+001	.0000
9.597+000	3.93037+001	.0000
9.596+000	3.93072+001	.0000
9.595+000	3.93107+001	.0000
9.594+000	3.93142+001	.0000
9.593+000	3.93177+001	.0000
9.592+000	3.93212+001	.0000
9.591+000	3.93247+001	.0000
9.590+000	3.93282+001	.0000
9.589+000	3.93317+001	.0000
9.588+000	3.93352+001	.0000
9.587+000	3.93387+001	.0000
9.586+000	3.93422+001	.0000
9.585+000	3.93457+001	.0000
9.584+000	3.93492+001	.0000
9.583+000	3.93527+001	.0000
9.582+000	3.93562+001	.0000
9.581+000	3.93597+001	.0000
9.580+000	3.93632+001	.0000
9.579+000	3.93667+001	.0000
9.578+000	3.93702+001	.0000
9.577+000	3.93737+001	.0000
9.576+000	3.93772+001	.0000
9.575+000	3.93807+001	.0000
9.574+000	3.93842+001	.0000
9.573+000	3.93877+001	.0000
9.572+000	3.93912+001	.0000
9.571+000	3.93947+001	.0000
9.570+000	3.93982+001	.0000
9.569+000	3.94017+001	.0000
9.568+000	3.94052+001	.0000
9.567+000	3.94087+001	.0000
9.566+000	3.94122+001	.0000
9.565+000	3.94157+001	.0000
9.564+000	3.94192+001	.0000
9.563+000	3.94227+001	.0000
9.562+000	3.94262+001	.0000
9.561+000	3.94297+001	.0000
9.560+000	3.94332+001	.0000
9.559+000	3.94367+001	.0000
9.558+000	3.94402+001	.0000
9.557+000	3.94437+001	.0000
9.556+000	3.94472+001	.0000
9.555+000	3.94507+001	.0000
9.554+000	3.94542+001	.0000
9.553+000	3.94577+001	.0000
9.552+000	3.94612+001	.0000
9.551+000	3.94647+001	.0000
9.550+000	3.94682+001	.0000
9.549+000	3.94717+001	.0000
9.548+000	3.94752+001	.0000
9.547+000	3.94787+001	.0000
9.546+000	3.94822+001	.0000
9.545+000	3.94857+001	.0000
9.544+000	3.94892+001	.0000
9.543+000	3.94927+001	.0000
9.542+000	3.94962+001	.0000
9.541+000	3.94997+001	.0000
9.540+000	3.95032+001	.0000
9.539+000	3.95067+001	.0000
9.538+000	3.95102+001	.0000
9.537+000	3.95137+001	.0000
9.536+000	3.95172+001	.0000
9.535+000	3.95207+001	.0000
9.534+000	3.95242+001	.0000
9.533+000	3.95277+001	.0000
9.532+000	3.95312+001	.0000
9.531+000	3.95347+001	.0000
9.530+000	3.95382+001	.0000
9.529+000	3.95417+001	.0000
9.528+000	3.95452+001	.0000
9.527+000	3.95487+001	.0000
9.526+000	3.95522+001	.0000
9.525+000	3.95557+001	.0000
9.524+000	3.95592+001	.0000
9.523+000	3.95627+001	.0000
9.522+000	3.95662+001	.0000
9.521+000	3.95697+001	.0000
9.520+000	3.95732+001	.0000
9.519+000	3.95767+001	.0000
9.518+000	3.95802+001	.0000
9.517+000	3.95837+001	.0000
9.516+000	3.95872+001	.0000
9.515+000	3.95907+001	.0000
9.514+000	3.95942+001	.0000
9.513+000	3.95977+001	.0000
9.512+000	3.96012+001	.0000
9.511+000	3.96047+001	.0000
9.510+000	3.96082+001	.0000
9.509+000	3.96117+001	.0000
9.508+000	3.96152+001	.0000
9.507+000	3.96187+001	.0000
9.506+000	3.96222+001	.0000
9.505+000	3.96257+001	.0000
9.504+000	3.96292+001	.0000
9.503+000	3.96327+001	.0000
9.502+000	3.96362+001	.0000
9.501+000	3.96397+001	.0000
9.500+000	3.96432+001	.0000
9.499+		

AD-A114 417

ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(II)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU

NL

6 . 6
5
2000-04-12

END
DATE
11-16-82
6 . 82
DTIG

TOTAL RADIANCE
 (MATTES M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET	BACKGROUND	CONTRAST RATIO
4.9751+000	2010	1.5362+000	1.6447+000	-6.5959-002
4.8309+000	2070	1.3232+000	1.4094+000	-6.1179-002
4.6948+000	2130	1.1211+000	1.2059+000	-7.0270-002
4.5662+000	2190	9.6050-001	1.0274+000	-6.5111-002
4.4444+000	2250	8.4575-001	8.5716-001	-7.4691-002
4.3239+000	2310	7.3211-001	7.3217-001	-7.7684-005
4.2194+000	2370	6.1542-001	6.1611-001	-1.1165-003
4.1152+000	2430	5.1886-001	5.3016-001	-7.1316-002
4.0161+000	2490	4.6102-001	4.5672-001	-9.4083-003
3.9216+000	2550	4.1435-001	3.9728-001	-4.2959-002
3.8314+000	2610	3.7419-001	3.4655-001	-7.9745-002
3.7453+000	2670	3.2256-001	2.8950-001	1.1420-001

DETECTOR-RESPONSE WAVELENGTH-INTEGRATED

(WATTS M-2 SR-1)

TARGET EMISSION	4.9117-002
TARGET REFLECTANCE	1.8157-003
PARTIAL ATMOSPHERIC EMISSION	4.3069-003
PARTIAL PATH RADIANCE	7.7355-005
TOTAL TARGET RADIANCE	9.4039-002
GROUND EMISSION	.0000
GROUND REFLECTANCE	.0000
TOTAL ATMOSPHERIC EMISSION	9.7352-002
TOTAL PATH RADIANCE	9.1328-004
TOTAL BACKGROUND RADIANCE	9.8265-002

CONTRAST -4.3004-002

DIRECT SUNLIGHT .0000
(WATTS M-2)

LT4M ATMOSPHERIC TRANSMISSION MODULE

PROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE

HORIZONTAL PATH, ALTITUDE = .002 KM, RANGE = .400 KM

MODEL ATMOSPHERE 4 = SUB-ARCTIC SUMMER

FREQUENCY RANGE V1= 2010.0 CM-1 TO V2= 2710.0 CM-1 FOR DV = 60.0 CM-1 < 3.69 - 4.98 MICRONS ,
EQUIVALENT SEA LEVEL ABSORBER AMOUNTS

WATER VAPOUR GM CM ⁻²	CO ₂ ETC. KM	OZONE ATM CM	NITROGEN (CONT) KM	H ₂ O (CONT) GM CM ⁻²	MOL KM	SCAT	OZONE(U-V) ATM CM
W(1-6,8)= W(10)=	.355+000 .372+000	.905-003	.295+000	.595-002 .546-001	.380+000		.915-003

	NITRIC ACID	S02	NH3	N02		
W(11-16)=	.000	.834-002	.779-002	.337-002	.433-002	.707-004

FREQ	WAVELENGTH	H2O	CO2+	OZONE	N2 C	H2O C	MOL S	NITRIC	SO2	HN03	NO2	INTEGRATED ABSORPTION	TOTAL	AEROSOL
CM-1	MICRONS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS
2010	4.9751	.6090	.9987	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	12.6982	.5767	.9488
2010	4.8309	.7608	.9529	.9996	1.0000	.8220	1.0000	1.0000	1.0000	1.0000	1.0000	.5648	.9420	.9420
1300	4.6948	.8649	.9853	.9990	.9990	.9066	1.0000	1.0000	1.0000	1.0000	1.0000	.5681	.9423	.9423
1300	4.5662	.9398	.9313	1.0000	.9915	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	.709	.9416	.9416
1300	4.4444	.9367	.9230	1.0000	.9807	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	.2813	.9408	.9408
1300	4.3290	.9769	.0011	1.0000	.9652	1.0000	.9791	1.0000	1.0000	1.0000	1.0000	.6153	.9400	.9400
1300	4.2194	.9925	.0172	1.0000	.9675	1.0000	.9848	1.0000	1.0000	1.0000	1.0000	.9006	.9400	.9400
1300	4.1152	.9977	.9941	1.0000	.9723	1.0000	.9890	1.0000	1.0000	1.0000	1.0000	.9181	.9400	.9400
1300	4.0161	.9998	.9941	1.0000	.9890	1.0000	.9890	1.0000	1.0000	1.0000	1.0000	.9070	.9400	.9400
1300	.9216	.9903	.9768	1.0000	.9972	1.0000	.9916	1.0000	1.0000	1.0000	1.0000	.262.2208	.8802	.9482
2670	3.7453	.9383	.9981	1.0000	.9996	.9916	1.0000	1.0000	1.0000	1.0000	1.0000			

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

INTEGRATED ASORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358

LT4M ATMOSPHERIC TRANSMISSION MODULE

PROGRAM WILL BE EXECUTED IN THE EMISSION MODE

HORIZONTAL PATH, ALTITUDE = .002 KM, RANGE = .400 KM

MODEL ATMOSPHERE 4 = SUB-ARCTIC SUMMER

FREQUENCY RANGE V1= 2010.0 CM-1 TO V2= 2710.0 CM-1 FOR DV = 60.0 CM-1 < 3.69 - 4.98 MICRONS >

EQUIVALENT SEA LEVEL ABSORBER AMOUNTS

	WATER VAPOUR GM CM-2	CO ₂ ETC. KM	OZONE ATM CM	NITROGEN (CONT) KM	H ₂ O (CONT) GM CM-2	MOL SCAT KM	OZONE(U-V) ATH CM
W(1-6,8)=	.355+000	.372+000	.905-003	.295+000	.595-002 .546-001	.380+000	.915-003
W(10)=							

	NITRIC ACID	S02	NH3	N02		
W(11-16)=	.000	.834-002	.779-002	.337-002	.433-002	.707-004

CUMULATIVE ABSORBER AMOUNTS FOR THE ATMOSPHERIC PATH

	H ₂ O	CO ₂ + .03	N ₂ .905-003	H ₂ O C .295+000	MOL S .595-002	O3 UV .380+000	H ₂ O C .915-003	HNO ₃ .546-001 .000	S02 .834-002	NH3 .779-002	N02 .337-002	TAVE 287.000
1	.355+000	.372+000	.905-003	.295+000	.595-002	.380+000	.915-003	.546-001 .000	.834-002	.779-002	.337-002	287.000

FR(CM-1)	WVL(MICRON)	RADIANCE(WATTS/CM2-STER-XXX) PER CM-1 PER MICRON	INTEGRAL	TRANS	--- AERO TRAN ---	RES	
2010.0	4.975124	.17216-006	.69555-004	.51648-005	.576727	.94820	.98942
2070.0	4.830918	.14311-006	.61320-004	.13751-004	.564830	.94820	.98942
2130.0	4.694836	.71276-007	.32337-004	.18028-004	.731250	.94820	.98942
2190.0	4.566210	.56657-007	.27173-004	.21427-004	.734479	.94820	.98942
2250.0	4.444444	.12310-006	.62320-004	.28813-004	.281325	.94820	.98942
2310.0	4.329004	.13708-006	.73146-004	.37038-004	.000967	.94820	.98942
2370.0	4.219409	.10801-006	.60666-004	.43518-004	.015339	.94820	.98942
2430.0	4.115226	.87026-008	.51388-005	.44040-004	.900563	.94820	.98942
2490.0	4.016064	.57098-008	.35402-005	.44383-004	.918083	.94820	.98942
2550.0	3.921569	.51549-008	.323530-005	.44692-004	.906379	.94820	.98942
2610.0	3.831418	.40774-008	.277776-005	.44937-004	.907301	.94820	.98942
2670.0	3.745318	.41768-008	.29776-005	.45188-004	.880173	.94820	.98942

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

RADMIN 2610.000 .40774-008
RADMAX 2010.000 .17216-006

INTEGRATED ASORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358
INTEGRATED RADIANCE = .45188-004 WATT CM -2 SR

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .4711-001

END EOSAEL RUN

ELECTRO-OPTICS DISTRIBUTION LIST

Commander
US Army Aviation School
Fort Rucker, AL 36362

Commander
US Army Aviation Center
ATTN: ATZQ-D-MA (Mr. Oliver N. Heath)
Fort Rucker, AL 36362

Commander
US Army Aviation Center
ATTN: ATZQ-D-MS (Mr. Donald Wagner)
Fort Rucker, AL 36362

NASA/Marshall Space Flight Center
ATTN: ES-83 (Otha H. Vaughan, Jr.)
Huntsville, AL 35812

NASA/Marshall Space Flight Center
Atmospheric Sciences Division
ATTN: Code ES-81 (Dr. William W. Vaughan)
Huntsville, AL 35812

Nichols Research Corporation
ATTN: Dr. Lary W. Pinkley
4040 South Memorial Parkway
Huntsville, AL 35802

John M. Hobbie
c/o Kentron International
2003 Byrd Spring Road
Huntsville, AL 35802

Mr. Ray Baker
Lockheed-Missile & Space Company
4800 Bradford Blvd
Huntsville, AL 35807

Commander
US Army Missile Command
ATTN: DRSMI-OG (Mr. Donald R. Peterson)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-OGA (Dr. Bruce W. Fowler)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REL (Dr. George Emmons)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Huey F. Anderson)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Mr. Maxwell W. Harper)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Mr. Gene Widenhofer)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RHC (Dr. Julius Q. Lilly)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
Redstone Scientific Information Center
ATTN: DRSMI-RPRD (Documents Section)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRA (Dr. Oskar Essenwanger)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRO (Mr. Charles Christensen)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRO (Dr. George A. Tanton)
Redstone Arsenal, AL 35809

Commander
US Army Communications Command
ATTN: CC-OPS-PP
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-CS (Mr. Richard G. Cundy)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-MD (Mr. Harry Wilder)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CS-C (2LT Coffman)
Fort Huachuca, AZ 85613

Commander
US Army Yuma Proving Ground
ATTN: STEYP-MSA-TL
Bldg 2105
Yuma, AZ 85364

Northrop Corporation
Electro-Mechanical Division
ATTN: Dr. Richard D. Tooley
500 East Orangethorpe Avenue
Anaheim, CA 92801

Commander
Naval Weapons Center
ATTN: Code 3918 (Dr. Alexis Shlanta)
China Lake, CA 93555

Hughes Helicopters
Army Advanced Attack Helicopter Weapons
ATTN: Mr. Charles R. Hill
Centinela and Teale Streets
Bldg 305, MS T-73A
Culter City, CA 90230

Commander
US Army Combat Developments
Experimentation Command
ATTN: ATEC-PL-M (Mr. Gary G. Love)
Fort Ord, CA 93941

SRI International
ATTN: K2060/Dr. Edward E. Uthe
333 Ravenswood Avenue
Menlo Park, CA 94025

SRI International
ATTN: Mr. J. E. Van der Laan
333 Ravenswood Avenue
Menlo Park, CA 94025

Joane May
Naval Environmental Prediction
Research Facility (NEPRF)
ATTN: Library
Monterey, CA 93940

Sylvania Systems Group,
Western Division
GTE Products Corporation
ATTN: Technical Reports Library
P.O. Box 205
Mountain View, CA 94042

Sylvania Systems Group
Western Division
GTE Products Corporation
ATTN: Mr. Lee W. Carrier
P.O. Box 188
Mountain View, CA 94042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3250-3 (R. de Violini)
Point Mugu, CA 93042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3253 (Terry E. Battalino)
Point Mugu, CA 93042

Effects Technology Inc.
ATTN: Mr. John D. Carlyle
5383 Hollister Avenue
Santa Barbara, CA 93111

Commander
Naval Ocean Systems Center
ATTN: Code 532 (Dr. Juergen Richter)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 5322 (Mr. Herbert G. Hughes)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 4473 (Tech Library)
San Diego, CA 92152

The RAND Corporation
ATTN: Ralph Huschke
1700 Main Street
Santa Monica, CA 90406

Particle Measuring Systems, Inc.
ATTN: Dr. Robert G. Knollenberg
1855 South 57th Court
Boulder, CO 80301

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: Library, R-51, Technical Reports
325 Broadway
Boulder, CO 80303

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: R45X3 (Dr. Vernon E. Derr)
Boulder, CO 80303

US Department of Commerce
National Telecommunications and
Information Administration
Institute for Telecommunication Sciences
ATTN: Code 1-3426 (Dr. Hans J. Liebe)
Boulder, CO 80303

AFATL/DLODL
Technical Library
Eglin AFB, FL 32542

Commanding Officer
Naval Training Equipment Center
ATTN: Technical Information Center
Orlando, FL 32813

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. Robert W. McMillan
Atlanta, GA 30332

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. James C. Wiltse
Atlanta, GA 30332

Commandant
US Army Infantry Center
ATTN: ATSH-CD-MS-E (Mr. Robert McKenna)
Fort Benning, GA 31805

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-CS
Fort Gordon, GA 30905

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-O
Fort Gordon, GA 30905

USAFTAC/DNE
ATTN: Mr. Charles Glauber
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DNDP (LTC Kit G. Cottrell)
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DOOF (MAJ Robert Wright)
Scott AFB, IL 62225

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-CAA-Q (Mr. H. Kent Pickett)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Robert DeKinder, Jr.)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Mr. Kent I. Johnson)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-WE (LTC Darrell Holland)
Fort Leavenworth, KS 66027

President
USAARENBD
ATTN: ATZK-AE-TA (Dr. Charles R. Leake)
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-MS
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-SD
Fort Knox, KY 40121

Aerodyne Research Inc.
ATTN: Dr. John F. Ebersole
Crosby Drive
Bedford, MA 01730

Commander
Air Force Geophysics Laboratory
ATTN: OPA (Dr. Robert W. Fenn)
Hanscom AFB, MA 01731

Commander
Air Force Geophysics Laboratory
ATTN: OPI (Dr. Robert A. McClatchey)
Hanscom AFB, MA 01731

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. T. J. Goblick, B-370
P.O. Box 73
Lexington, MA 02173

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. Michael Gruber
P.O. Box 73
Lexington, MA 02173

Raytheon Company
Equipment Division
ATTN: Dr. Charles M. Sonnenschein
430 Boston Post Road
Wayland, MA 01778

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-BLB (Mr. Richard McGee)
Aberdeen Proving Ground, MD 21005

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Dr. Edward Stuebing)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Mr. Joseph Vervier)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLY-A (Mr. Ronald Pennsyle)
Aberdeen Proving Ground, MD 21010

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-TSB-S (STINFO)
Aberdeen Proving Ground, MD 21005

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CCM (W. H. Pepper)
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CG/DRDEL-DC/DRDEL-CS
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CT
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-PAO (M. Singleton)
2800 Powder Mill Road
Adelphi, MD 20783

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK
(Dr. Anthony Van de Wal, Jr.)
Aberdeen Proving Ground, MD 21005

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK-T (Mr. Sidney Gerard)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSSTE-AD-M (Mr. Warren M. Baity)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSSTE-AD-M (Dr. Norman E. Pentz)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-AAM (Mr. William Smith)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-CS (Mr. Philip H. Beavers)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GB (Wilbur L. Warfield)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GP (Mr. Fred Campbell)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GP (H. Stamper)
Aberdeen Proving Grounds, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GS
(Mr. Michael Starks/Mr. Julian Chernick)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-J (Mr James F. O'Bryon)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-LM (Mr. Robert M. Marchetti)
Aberdeen Proving Ground, MD 21005

Commander
Harry Diamond Laboratories
ATTN: Dr. William W. Carter
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM (Mr. Robert McCoskey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Robert Humphrey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Z. G. Sztankay)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Joseph Nemarich)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Air Force Systems Command
ATTN: WER (Mr. Richard F. Picano)
Andrews AFB, MD 20334

Martin Marietta Laboratories
ATTN: Jar Mo Chen
1450 South Rolling Road
Baltimore, MD 21227

Commander
US Army Concepts Analysis Agency
ATTN: CSCA-SMC (Mr. Hal E. Hock)
8120 Woodmont Avenue
Bethesda, MD 20014

Dr. A. D. Belmont
Research Division
Control Data Corporation
P.O. Box 1249
Minneapolis, MN 55440

Director
National Security Agency
ATTN: R52/Dr. Douglas Woods
Fort George G. Meade, MD 20755

Director
US Army Engr Waterways Experiment Station
ATTN: WESEN (Mr. James Mason)
P.O. Box 631
Vicksburg, MS 39180

Chief
Intelligence Materiel Development
& Support Office
US Army Electronic Warfare Laboratory
ATTN: DELEW-I (LTC Kenneth E. Thomas)
Fort George G. Meade, MD 20755

Dr. Jerry Davis
Department of Marine, Earth
and Atmospheric Sciences
North Carolina State University
Raleigh, NC 27650

The Johns Hopkins University
Applied Physics Laboratory
ATTN: Dr. Michael J. Lun
John Hopkins Road
Laurel, MD 20810

Commander
US Army Research Office
ATTN: DRXRO-GS (Dr. Leo Alpert)
P.O. Box 12211
Research Triangle Park, NC 27709

Dr. Stephen T. Hanley
1720 Rhodesia Avenue
Oxon Hill, MD 20022

Commander
US Army Research Office
ATTN: DRXRO-PP (Brenda Mann)
P.O. Box 12211
Research Triangle Park, NC 27709

Science Applications Inc.
ATTN: Mr. G. D. Currie
15 Research Drive
Ann Arbor, MI 48103

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RD (Dr. K. F. Sterrett)
Hanover, NH 03755

Science Applications Inc.
ATTN: Dr. Robert E. Turner
15 Research Drive
Ann Arbor, MI 48103

Commander
US Army Tank-Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Harry Young)
Warren, MI 48090

Commander/Director
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. George Aitken)
Hanover, NH 03755

Commander
US Army Tank Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Wallace Mick, Jr.)
Warren, MI 48090

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. Roger H. Berger)
Hanover, NH 03755

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-AC (Mr. James Greenfield)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-TSS (Bldg #59)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. Peteris Jansons)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. G. H. Waldron)
Dover, NJ 07801

Deputy Joint Project Manager
for Navy/USMC SAL GP
ATTN: DRCPM-CAWS-NV (CPT Joseph Micelli)
Dover, NJ 07801

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-I (Mr. David Longinotti)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-PE (Mr. Ben A. Di Campli)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-R-S (Mr. Donald L. Fofani)
Fort Monmouth, NJ 07703

Director
US Army Electronics Technology &
Devices Laboratory
ATTN: DELET-DD (S. Danko)
Fort Monmouth, NJ 07703

Project Manager
FIREFINDER/REMBASS
ATTN: DRCPM-FFR-TM (Mr. John M. Bialo)
Fort Monmouth, NJ 07703

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-SA (Dr. Walter S. McAfee)
Fort Monmouth, NJ 07703

OLA, 2WS (MAC)
Holloman AFB, NM 88330
Commander
Air Force Weapons Laboratory
ATTN: AFWL/WE (MAJ John R. Elrick)
Kirtland, AFB, NM 87117

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL (Dolores Anguiano)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. Louie Dominguez)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. William J. Leach)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TGP (Mr. Roger F. Willis)
White Sands Missile Range, NM 88002

Director
Office of Missle Electronic Warfare
ATTN: DELEW-M-STO (Dr. Steven Kovel)
White Sands Missile Range, NM 88002

Office of the Test Director
Joint Services EO GW CM Test Program
ATTN: DRXDE-TD (Mr. Weldon Findley)
White Sands Missile Range, NM 88002

Commander
US Army White Sands Missile Range
ATTN: STEWS-PT-AL (Laurel B. Saunders)
White Sands Missile Range, NM 88002

Commander
US Army R&D Coordinator
US Embassy - Bonn
Box 165
APO New York 09080

Grumman Aerospace Corporation
Research Department - MS A08-35
ATTN: John E. A. Selby
Bethpage, NY 11714

Rome Air Development Center
ATTN: Documents Library
TSLD (Bette Smith)
Griffiss AFB, NY 13441

Dr. Roberto Vaglio-Laurin
Faculty of Arts and Science
Dept. of Applied Science
26-36 Stuyvesant Street
New York, NY 10003

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (Mr. Harold Geltmacher)
Wright-Patterson AFB, OH 45433

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (CPT William C. Smith)
Wright-Patterson AFB, OH 45433

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R (CPT James M. Watson)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CD-MS
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: NOAA Liaison Officer
(CDR Jeffrey G. Carlen)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
Morris Swett Library
ATTN: Reference Librarian
Fort Sill, OK 73503

Commander
Naval Air Development Center
ATTN: Code 301 (Mr. George F. Eck)
Warminster, PA 18974

The University of Texas at El Paso
Electrical Engineering Department
ATTN: Dr. Joseph H. Pierluissi
El Paso, TX 79968

Commandant
US Army Air Defense School
ATTN: ATSA-CD-SC-A (CPT Charles T. Thorn)
Fort Bliss, TX 79916

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-OP-Q (CPT Henry C. Cobb, Jr.)
Fort Hood, TX 76544

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-SCI (Dr. Darrell W. Collier)
Fort Hood, TX 76544

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-L
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-M (Mr. Paul E. Carlson)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. John Trethewey)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. William Peterson)
Dugway, UT 84022

Defense Documentation Center
ATTN: DDC-TCA
Cameron Station Bldg 5
Alexandria, VA 22314
12

Ballistic Missile Defense Program Office
ATTN: DACS-BMT (Colonel Harry F. Ennis)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Technical Information Center
ATTN: DDA-2 (Mr. James E. Shafer)
Cameron Station, Bldg 5
Alexandria, VA 22314

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCBSI-EE (Mr. Albert Giambalvo)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCLDC (Mr. James Bender)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Advanced Rsch Projects Agency
ATTN: Steve Zakanyez
1400 Wilson Blvd
Arlington, VA 22209

Defense Advanced Rsch Projects Agency
ATTN: Dr. James Tegnelia
1400 Wilson Blvd
Arlington, VA 22209

Institute for Defense Analyses
ATTN: Mr. Lucien M. Biberman
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Ernest Bauer
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Hans G. Wolfhard
400 Army-Navy Drive
Arlington, VA 22202

System Planning Corporation
ATTN: Mr. Daniel Friedman
1500 Wilson Boulevard
Arlington, VA 22209

System Planning Corporation
ATTN: COL Hank Shelton
1500 Wilson Boulevard
Arlington, VA 22209

US Army Intelligence & Security Command
ATTN: Edwin Speakman, Scientific Advisor
Arlington Hall Station
Arlington, VA 22212

Commander
US Army Operational Test
& Evaluation Agency
ATTN: CSTE-ED (Mr. Floyd I. Hill)
5600 Columbia Pike
Falls Church, VA 22041

Commander and Director
US Army Engineer Topographic Laboratories
ATTN: ETL-GS-A (Mr. Thomas Neidringhaus)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Rudolf G. Buser)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Robert S. Rodhe)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Joseph R. Moulton)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Luanne P. Obert)
Fort Belvoir, VA 22060

Director
US Army Night Vision & Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Thomas W. Cassidy)
Fort Belvoir, VA 22060

Director
US Army Night Vision & Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Richard J. Bergemann)
Fort Belvoir, VA 22060

Director
US Army Night Vision & Electro-Optics Laboratory
ATTN: DELNV-VI (Dr. James A. Ratches)
Fort Belvoir, VA 22060

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN-M
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-F-A (Mr. Chris O'Connor, Jr.)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-IE-R (Mr. David M. Ingram)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-M-I/ATCD-M-A
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATDOC-TA (Dr. Marvin P. Pastel)
Fort Monroe, VA 23651

Department of the Air Force
OL-I, AWS
Fort Monroe, VA 23651

Department of the Air Force
HQ 5 Weather Wing (MAC)
ATTN: 5 WW/DN
Langley Air Force Base, VA 23655

Commander
US Army INSCOM/Quest Research Corporation
ATTN: Mr. Donald Wilmot
6845 Elm Street, Suite 407
McLean, VA 22101

General Research Corporation
ATTN: Dr. Ralph Zirkind
7655 Old Springhouse Road
McLean, VA 22102

Science Applications, Inc.
8400 Westpark Drive
ATTN: Dr. John E. Cockayne
McLean, VA 22102

US Army Nuclear & Chemical Agency
ATTN: MONA-WE (Dr. John A. Berberet)
7500 Backlick Road, Bldg 2073
Springfield, VA 22150

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-EA (Mr. Douglas Harkleroad)
Vint Hill Farms Station
Warrenton, VA 22186

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-OS (Dr. Royal H. Burkhardt)
Vint Hill Farms Station
Warrenton, VA 22186

Commander
US Army Cold Regions Test Center
ATTN: STECR-TD (Mr. Jerold Barger)
APO Seattle, WA 98733

HQDA (SAUS-OR/Hunter M. Woodall, Jr./
Dr. Herbert K. Fallin)
Rm 2E 614, Pentagon
Washington, DC 20301

COL Elbert W. Friday, Jr.
OUSDRE
Rm 3D 129, Pentagon
Washington, DC 20301

Defense Communications Agency
Technical Library Center
Code 222
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: Technical Library (Mrs. Betty Fox)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: RAAE (Dr. Carl Fitz)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: SPAS (Mr. Donald J. Kohler)
Washington, DC 20305

Defense Intelligence Agency
ATTN: DT/AC (LTC Robert Poplawski)
Washington, DC 20301

HQDA (DAMA-ARZ-D/Dr. Verderame)
Washington, DC 20310

HQDA (DAMI-ISP/Mr. Beck)
Washington, DC 20310

Department of the Army
Deputy Chief of Staff for
Operations and Plans
ATTN: DAMO-RQ
Washington, DC 20310

Department of the Army
Director of Telecommunications and
Command and Control
ATTN: DAMO-TCZ
Washington, DC 20310

Department of the Army
Assistant Chief of Staff for Intelligence
ATTN: DAMI-TS
Washington, DC 20310

HQDA (DAEN-RDM/Dr. de Percin)
Casimir Pulaski Building
20 Massachusetts Avenue
Room 6203
Washington, DC 20314

National Science Foundation
Division of Atmospheric Sciences
ATTN: Dr. Eugene W. Bierly
1800 G. Street, N.W.
Washington, DC 20550

Director
Naval Research Laboratory
ATTN: Code 4320 (Dr. Lothar H. Ruhnke)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6009 (Dr. John MacCallum, Jr.)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6530 (Mr. Raymond A. Patten)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6533 (Dr. James A. Dowling)
Washington, DC 20375

ATMOSPHERIC SCIENCES RESEARCH REPORTS

1. Lindberg, J. D. "An Improvement to a Method for Measuring the Absorption Coefficient of Atmospheric Dust and other Strongly Absorbing Powders," ECOM-5565, July 1975.
2. Avara, Elton P., "Mesoscale Wind Shears Derived from Thermal Winds," ECOM-5566, July 1975.
3. Gomez, Richard B., and Joseph H. Pierluissi, "Incomplete Gamma Function Approximation for King's Strong-Line Transmittance Model," ECOM-5567, July 1975.
4. Blanco, A. J., and B. F. Engebos, "Ballistic Wind Weighting Functions for Tank Projectiles," ECOM-5568, August 1975.
5. Taylor, Fredrick J., Jack Smith, and Thomas H. Pries, "Crosswind Measurements through Pattern Recognition Techniques," ECOM-5569, July 1975.
6. Walters, D. L., "Crosswind Weighting Functions for Direct-Fire Projectiles," ECOM-5570, August 1975.
7. Duncan, Louis D., "An Improved Algorithm for the Iterated Minimal Information Solution for Remote Sounding of Temperature," ECOM-5571, August 1975.
8. Robbiani, Raymond L., "Tactical Field Demonstration of Mobile Weather Radar Set AN/TPS-41 at Fort Rucker, Alabama," ECOM-5572, August 1975.
9. Miers, B., G. Blackman, D. Langer, and N. Lorimier, "Analysis of SMS/GOES Film Data," ECOM-5573, September 1975.
10. Manquero, Carlos, Louis Duncan, and Rufus Bruce, "An Indication from Satellite Measurements of Atmospheric CO₂ Variability," ECOM-5574, September 1975.
11. Petracca, Carmine, and James D. Lindberg, "Installation and Operation of an Atmospheric Particulate Collector," ECOM-5575, September 1975.
12. Avara, Elton P., and George Alexander, "Empirical Investigation of Three Iterative Methods for Inverting the Radiative Transfer Equation," ECOM-5576, October 1975.
13. Alexander, George D., "A Digital Data Acquisition Interface for the SMS Direct Readout Ground Station - Concept and Preliminary Design," ECOM-5577, October 1975.
14. Cantor, Israel, "Enhancement of Point Source Thermal Radiation Under Clouds in a Nonattenuating Medium," ECOM-5578, October 1975.

15. Norton, Colburn, and Glenn Hoidal, "The Diurnal Variation of Mixing Height by Month over White Sands Missile Range, NM," ECOM-5579, November 1975.
16. Avara, Elton P., "On the Spectrum Analysis of Binary Data," ECOM-5580, November 1975.
17. Taylor, Fredrick J., Thomas H. Pries, and Chao-Huan Huang, "Optimal Wind Velocity Estimation," ECOM-5581, December 1975.
18. Avara, Elton P., "Some Effects of Autocorrelated and Cross-Correlated Noise on the Analysis of Variance," ECOM-5582, December 1975.
19. Gillespie, Patti S., R. L. Armstrong, and Kenneth O. White, "The Spectral Characteristics and Atmospheric CO₂ Absorption of the Ho⁺³:YLF Laser at 2.05μm," ECOM-5583, December 1975.
20. Novlan, David J., "An Empirical Method of Forecasting Thunderstorms for the White Sands Missile Range," ECOM-5584, February 1976.
21. Avara, Elton P., "Randomization Effects in Hypothesis Testing with Autocorrelated Noise," ECOM-5585, February 1976.
22. Watkins, Wendell R., "Improvements in Long Path Absorption Cell Measurement," ECOM-5586, March 1976.
23. Thomas, Joe, George D. Alexander, and Marvin Dubbin, "SATTEL - An Army Dedicated Meteorological Telemetry System," ECOM-5587, March 1976.
24. Kennedy, Bruce W., and Delbert Bynum, "Army User Test Program for the RDT&E-XM-75 Meteorological Rocket," ECOM-5588, April 1976.
25. Barnett, Kenneth M., "A Description of the Artillery Meteorological Comparisons at White Sands Missile Range, October 1974 - December 1974 ('PASS' - Prototype Artillery [Meteorological] Subsystem)," ECOM-5589, April 1976.
26. Miller, Walter B., "Preliminary Analysis of Fall-of-Shot From Project 'PASS'," ECOM-5590, April 1976.
27. Avara, Elton P., "Error Analysis of Minimum Information and Smith's Direct Methods for Inverting the Radiative Transfer Equation," ECOM-5591, April 1976.
28. Yee, Young P., James D. Horn, and George Alexander, "Synoptic Thermal Wind Calculations from Radiosonde Observations Over the Southwestern United States," ECOM-5592, May 1976.

29. Duncan, Louis D., and Mary Ann Seagraves, "Applications of Empirical Corrections to NOAA-4 VTPR Observations," ECOM-5593, May 1976.
30. Miers, Bruce T., and Steve Weaver, "Applications of Meteorological Satellite Data to Weather Sensitive Army Operations," ECOM-5594, May 1976.
31. Sharenow, Moses, "Redesign and Improvement of Balloon ML-566," ECOM-5595, June 1976.
32. Hansen, Frank V., "The Depth of the Surface Boundary Layer," ECOM-5596, June 1976.
33. Pinnick, R. G., and E. B. Stenmark, "Response Calculations for a Commerical Light-Scattering Aerosol Counter," ECOM-5597, July 1976.
34. Mason, J., and G. B. Hoidal, "Visibility as an Estimator of Infrared Transmittance," ECOM-5598, July 1976.
35. Bruce, Rufus E., Louis D. Duncan, and Joseph H. Pierluissi, "Experimental Study of the Relationship Between Radiosonde Temperatures and Radiometric-Area Temperatures," ECOM-5599, August 1976.
36. Duncan, Louis D., "Stratospheric Wind Shear Computed from Satellite Thermal Sounder Measurements," ECOM-5800, September 1976.
37. Taylor, F., P. Mohan, P. Joseph, and T. Pries, "An All Digital Automated Wind Measurement System," ECOM-5801, September 1976.
38. Bruce, Charles, "Development of Spectrophones for CW and Pulsed Radiation Sources," ECOM-5802, September 1976.
39. Duncan, Louis D., and Mary Ann Seagraves, "Another Method for Estimating Clear Column Radiances," ECOM-5803, October 1976.
40. Blanco, Abel J., and Larry E. Taylor, "Artillery Meteorological Analysis of Project Pass," ECOM-5804, October 1976.
41. Miller, Walter, and Bernard Engebos, "A Mathematical Structure for Refinement of Sound Ranging Estimates," ECOM-5805, November 1976.
42. Gillespie, James B., and James D. Lindberg, "A Method to Obtain Diffuse Reflectance Measurements from 1.0 and 3.0 μ m Using a Cary 17I Spectrophotometer," ECOM-5806, November 1976.
43. Rubio, Roberto, and Robert O. Olsen, "A Study of the Effects of Temperature Variations on Radio Wave Absorption," ECOM-5807, November 1976.

44. Ballard, Harold N., "Temperature Measurements in the Stratosphere from Balloon-Borne Instrument Platforms, 1968-1975," ECOM-5808, December 1976.
45. Monahan, H. H., "An Approach to the Short-Range Prediction of Early Morning Radiation Fog," ECOM-5809, January 1977.
46. Engebos, Bernard Francis, "Introduction to Multiple State Multiple Action Decision Theory and Its Relation to Mixing Structures," ECOM-5810, January 1977.
47. Low, Richard D. H., "Effects of Cloud Particles on Remote Sensing from Space in the 10-Micrometer Infrared Region," ECOM-5811, January 1977.
48. Bonner, Robert S., and R. Newton, "Application of the AN/GVS-5 Laser Rangefinder to Cloud Base Height Measurements," ECOM-5812, February 1977.
49. Rubio, Roberto, "Lidar Detection of Subvisible Reentry Vehicle Erosive Atmospheric Material," ECOM-5813, March 1977.
50. Low, Richard D. H., and J. D. Horn, "Mesoscale Determination of Cloud-Top Height: Problems and Solutions," ECOM-5814, March 1977.
51. Duncan, Louis D., and Mary Ann Seagraves, "Evaluation of the NOAA-4 VTPR Thermal Winds for Nuclear Fallout Predictions," ECOM-5815, March 1977.
52. Randhawa, Jagir S., M. Izquierdo, Carlos McDonald, and Zvi Salpeter, "Stratospheric Ozone Density as Measured by a Chemiluminescent Sensor During the Stratcom VI-A Flight," ECOM-5816, April 1977.
53. Rubio, Roberto, and Mike Izquierdo, "Measurements of Net Atmospheric Irradiance in the 0.7- to 2.8-Micrometer Infrared Region," ECOM-5817, May 1977.
54. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, Consultant for Chemical Kinetics, "Calculation of Selected Atmospheric Composition Parameters for the Mid-Latitude, September Stratosphere," ECOM-5818, May 1977.
55. Mitchell, J. D., R. S. Sagar, and R. O. Olsen, "Positive Ions in the Middle Atmosphere During Sunrise Conditions," ECOM-5819, May 1977.
56. White, Kenneth O., Wendell R. Watkins, Stuart A. Schleusener, and Ronald L. Johnson, "Solid-State Laser Wavelength Identification Using a Reference Absorber," ECOM-5820, June 1977.
57. Watkins, Wendell R., and Richard G. Dixon, "Automation of Long-Path Absorption Cell Measurements," ECOM-5821, June 1977.

58. Taylor, S. E., J. M. Davis, and J. B. Mason, "Analysis of Observed Soil Skin Moisture Effects on Reflectance," ECOM-5822, June 1977.
59. Duncan, Louis D., and Mary Ann Seagraves, "Fallout Predictions Computed from Satellite Derived Winds," ECOM-5823, June 1977.
60. Snider, D. E., D. G. Murcray, F. H. Murcray, and W. J. Williams, "Investigation of High-Altitude Enhanced Infrared Background Emissions," (U), SECRET, ECOM-5824, June 1977.
61. Dubbin, Marvin H., and Dennis Hall, "Synchronous Meteorological Satellite Direct Readout Ground System Digital Video Electronics," ECOM-5825, June 1977.
62. Miller, W., and B. Engebos, "A Preliminary Analysis of Two Sound Ranging Algorithms," ECOM-5826, July 1977.
63. Kennedy, Bruce W., and James K. Luers, "Ballistic Sphere Techniques for Measuring Atmospheric Parameters," ECOM-5827, July 1977.
64. Duncan, Louis D., "Zenith Angle Variation of Satellite Thermal Sounder Measurements," ECOM-5828, August 1977.
65. Hansen, Frank V., "The Critical Richardson Number," ECOM-5829, September 1977.
66. Ballard, Harold N., and Frank P. Hudson (Compilers), "Stratospheric Composition Balloon-Borne Experiment," ECOM-5830, October 1977.
67. Barr, William C., and Arnold C. Peterson, "Wind Measuring Accuracy Test of Meteorological Systems," ECOM-5831, November 1977.
68. Ethridge, G. A., and F. V. Hansen, "Atmospheric Diffusion: Similarity Theory and Empirical Derivations for Use in Boundary Layer Diffusion Problems," ECOM-5832, November 1977.
69. Low, Richard D. H., "The Internal Cloud Radiation Field and a Technique for Determining Cloud Blackness," ECOM-5833, December 1977.
70. Watkins, Wendell R., Kenneth O. White, Charles W. Bruce, Donald L. Walters, and James D. Lindberg, "Measurements Required for Prediction of High Energy Laser Transmission," ECOM-5834, December 1977.
71. Rubio, Robert, "Investigation of Abrupt Decreases in Atmospherically Backscattered Laser Energy," ECOM-5835, December 1977.
72. Monahan, H. H., and R. M. Cionco, "An Interpretative Review of Existing Capabilities for Measuring and Forecasting Selected Weather Variables (Emphasizing Remote Means)," ASL-TR-0001, January 1978.

73. Heaps, Melvin G., "The 1979 Solar Eclipse and Validation of D-Region Models," ASL-TR-0002, March 1978.
74. Jennings, S. G., and J. B. Gillespie, "M.I.E. Theory Sensitivity Studies - The Effects of Aerosol Complex Refractive Index and Size Distribution Variations on Extinction and Absorption Coefficients, Part II: Analysis of the Computational Results," ASL-TR-0003, March 1978.
75. White, Kenneth O., et al, "Water Vapor Continuum Absorption in the $3.5\mu\text{m}$ to $4.0\mu\text{m}$ Region," ASL-TR-0004, March 1978.
76. Olsen, Robert O., and Bruce W. Kennedy, "ABRES Pretest Atmospheric Measurements," ASL-TR-0005, April 1978.
77. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, "Calculation of Atmospheric Composition in the High Latitude September Stratosphere," ASL-TR-0006, May 1978.
78. Watkins, Wendell R., et al, "Water Vapor Absorption Coefficients at HF Laser Wavelengths," ASL-TR-0007, May 1978.
79. Hansen, Frank V., "The Growth and Prediction of Nocturnal Inversions," ASL-TR-0008, May 1978.
80. Samuel, Christine, Charles Bruce, and Ralph Brewer, "Spectrophone Analysis of Gas Samples Obtained at Field Site," ASL-TR-0009, June 1978.
81. Pinnick, R. G., et al., "Vertical Structure in Atmospheric Fog and Haze and its Effects on IR Extinction," ASL-TR-0010, July 1978.
82. Low, Richard D. H., Louis D. Duncan, and Richard B. Gomez, "The Microphysical Basis of Fog Optical Characterization," ASL-TR-0011, August 1978.
83. Heaps, Melvin G., "The Effect of a Solar Proton Event on the Minor Neutral Constituents of the Summer Polar Mesosphere," ASL-TR-0012, August 1978.
84. Mason, James B., "Light Attenuation in Falling Snow," ASL-TR-0013, August 1978.
85. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' Meteorological Application," ASL-TR-0014, September 1978.
86. Heaps, M. G., and F. E. Niles, "Modeling of Ion Chemistry of the D-Region: A Case Study Based Upon the 1966 Total Solar Eclipse," ASL-TR-0015, September 1978.

87. Jennings, S. G., and R. G. Pinnick, "Effects of Particulate Complex Refractive Index and Particle Size Distribution Variations on Atmospheric Extinction and Absorption for Visible Through Middle-Infrared Wavelengths," ASL-TR-0016, September 1978.
88. Watkins, Wendell R., Kenneth O. White, Lanny R. Bower, and Brian Z. Sojka, "Pressure Dependence of the Water Vapor Continuum Absorption in the 3.5- to 4.0-Micrometer Region," ASL-TR-0017, September 1978.
89. Miller, W. B., and B. F. Engebos, "Behavior of Four Sound Ranging Techniques in an Idealized Physical Environment," ASL-TR-0018, September 1978.
90. Gomez, Richard G., "Effectiveness Studies of the CBU-88/B Bomb, Cluster, Smoke Weapon," (U), CONFIDENTIAL ASL-TR-0019, September 1978.
91. Miller, August, Richard C. Shirkey, and Mary Ann Seagraves, "Calculation of Thermal Emission from Aerosols Using the Doubling Technique," ASL-TR-0020, November 1978.
92. Lindberg, James D., et al, "Measured Effects of Battlefield Dust and Smoke on Visible, Infrared, and Millimeter Wavelengths Propagation: A Preliminary Report on Dusty Infrared Test-I (DIRT-I)," ASL-TR-0021, January 1979.
93. Kennedy, Bruce W., Arthur Kinghorn, and B. R. Hixon, "Engineering Flight Tests of Range Meteorological Sounding System Radiosonde," ASL-TR-0022, February 1979.
94. Rubio, Roberto, and Don Hoock, "Microwave Effective Earth Radius Factor Variability at Wiesbaden and Balboa," ASL-TR-0023, February 1979.
95. Low, Richard D. H., "A Theoretical Investigation of Cloud/Fog Optical Properties and Their Spectral Correlations," ASL-TR-0024, February 1979.
96. Pinnick, R. G., and H. J. Auvermann, "Response Characteristics of Knollenberg Light-Scattering Aerosol Counters," ASL-TR-0025, February 1979.
97. Heaps, Melvin G., Robert O. Olsen, and Warren W. Berning, "Solar Eclipse 1979, Atmospheric Sciences Laboratory Program Overview," ASL-TR-0026, February 1979.
98. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' GR-8 Sound Ranging Data," ASL-TR-0027, March 1979.
99. Kennedy, Bruce W., and Jose M. Serna, "Meteorological Rocket Network System Reliability," ASL-TR-0028, March 1979.

100. Swingle, Donald M., "Effects of Arrival Time Errors in Weighted Range Equation Solutions for Linear Base Sound Ranging," ASL-TR-0029, April 1979.
101. Umstead, Robert K., Ricardo Pena, and Frank V. Hansen, "KWIK: An Algorithm for Calculating Munition Expenditures for Smoke Screening/Obscuration in Tactical Situations," ASL-TR-0030, April 1979.
102. D'Arcy, Edward M., "Accuracy Validation of the Modified Nike Hercules Radar," ASL-TR-0031, May 1979.
103. Rodriguez, Ruben, "Evaluation of the Passive Remote Crosswind Sensor," ASL-TR-0032, May 1979.
104. Barber, T. L., and R. Rodriguez, "Transit Time Lidar Measurement of Near-Surface Winds in the Atmosphere," ASL-TR-0033, May 1979.
105. Low, Richard D. H., Louis D. Duncan, and Y. Y. Roger R. Hsiao, "Micro-physical and Optical Properties of California Coastal Fogs at Fort Ord," ASL-TR-0034, June 1979.
106. Rodriguez, Ruben, and William J. Vechione, "Evaluation of the Saturation Resistant Crosswind Sensor," ASL-TR-0035, July 1979.
107. Ohmstede, William D., "The Dynamics of Material Layers," ASL-TR-0036, July 1979.
108. Pinnick, R. G., S. G. Jennings, Petr Chylek, and H. J. Auermann, "Relationships between IR Extinction Absorption, and Liquid Water Content of Fogs," ASL-TR-0037, August 1979.
109. Rodriguez, Ruben, and William J. Vechione, "Performance Evaluation of the Optical Crosswind Profiler," ASL-TR-0038, August 1979.
110. Miers, Bruce T., "Precipitation Estimation Using Satellite Data," ASL-TR-0039, September 1979.
111. Dickson, David H., and Charles M. Sonnenschein, "Helicopter Remote Wind Sensor System Description," ASL-TR-0040, September 1979.
112. Heaps, Melvin G., and Joseph M. Heimerl, "Validation of the Dairchem Code, I: Quiet Midlatitude Conditions," ASL-TR-0041, September 1979.
113. Bonner, Robert S., and William J. Lentz, "The Visioceilometer: A Portable Cloud Height and Visibility Indicator," ASL-TR-0042, October 1979.
114. Cohn, Stephen L., "The Role of Atmospheric Sulfates in Battlefield Obscurations," ASL-TR-0043, October 1979.

115. Fawbush, E. J., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF), White Sands Missile Range, New Mexico, Part I, 24 March to 8 April 1977," ASL-TR-0044, November 1979.
116. Barber, Ted L., "Short-Time Mass Variation in Natural Atmospheric Dust," ASL-TR-0045, November 1979.
117. Low, Richard D. H., "Fog Evolution in the Visible and Infrared Spectral Regions and its Meaning in Optical Modeling," ASL-TR-0046, December 1979.
118. Duncan, Louis D., et al, "The Electro-Optical Systems Atmospheric Effects Library, Volume I: Technical Documentation," ASL-TR-0047, December 1979.
119. Shirkey, R. C., et al, "Interim E-O SAEL, Volume II, Users Manual," ASL-TR-0048, December 1979.
120. Kobayashi, H. K., "Atmospheric Effects on Millimeter Radio Waves," ASL-TR-0049, January 1980.
121. Seagraves, Mary Ann, and Louis D. Duncan, "An Analysis of Transmittances Measured Through Battlefield Dust Clouds," ASL-TR-0050, February 1980.
122. Dickson, David H., and Jon E. Ottesen, "Helicopter Remote Wind Sensor Flight Test," ASL-TR-0051, February 1980.
123. Pinnick, R. G., and S. G. Jennings, "Relationships Between Radiative Properties and Mass Content of Phosphoric Acid, HC, Petroleum Oil, and Sulfuric Acid Military Smokes," ASL-TR-0052, April 1980.
124. Hinds, B. D., and J. B. Gillespie, "Optical Characterization of Atmospheric Particulates on San Nicolas Island, California," ASL-TR-0053, April 1980.
125. Miers, Bruce T., "Precipitation Estimation for Military Hydrology," ASL-TR-0054, April 1980.
126. Stenmark, Ernest B., "Objective Quality Control of Artillery Computer Meteorological Messages," ASL-TR-0055, April 1980.
127. Duncan, Louis D., and Richard D. H. Low, "Bimodal Size Distribution Models for Fogs at Meppen, Germany," ASL-TR-0056, April 1980.
128. Olsen, Robert O., and Jagir S. Randhawa, "The Influence of Atmospheric Dynamics on Ozone and Temperature Structure," ASL-TR-0057, May 1980.

129. Kennedy, Bruce W., et al, "Dusty Infrared Test-II (DIRT-II) Program," ASL-TR-0058, May 1980.
130. Heaps, Melvin G., Robert O. Olsen, Warren Berning, John Cross, and Arthur Gilcrease, "1979 Solar Eclipse, Part I - Atmospheric Sciences Laboratory Field Program Summary," ASL-TR-0059, May 1980
131. Miller, Walter B., "User's Guide for Passive Target Acquisition Program Two (PTAP-2)," ASL-TR-0060, June 1980.
132. Holt, E. H., editor, "Atmospheric Data Requirements for Battlefield Obscuration Applications," ASL-TR-0061, June 1980.
133. Shirkey, Richard C., August Miller, George H. Goedecke, and Yugal Behl, "Single Scattering Code AGAUSX: Theory, Applications, Comparisons, and Listing," ASL-TR-0052, July 1980.
134. Sojka, Brian Z., and Kenneth D. White, "Evaluation of Specialized Photoacoustic Absorption Chambers for Near-Millimeter Wave (NMMW) Propagation Measurements," ASL-TR-0063, August 1980.
135. Bruce, Charles W., Young Paul Yee, and S. G. Jennings, "In Situ Measurement of the Ratio of Aerosol Absorption to Extinction Coefficient," ASL-TR-0064, August 1980.
136. Yee, Young Paul, Charles W. Bruce, and Ralph J. Brewer, "Gaseous/Particulate Absorption Studies at WSMR using Laser Sourced Spectrophones," ASL-TR-0065, June 1980.
137. Lindberg, James D., Radon B. Loveland, Melvin Heaps, James B. Gillespie, and Andrew F. Lewis, "Battlefield Dust and Atmospheric Characterization Measurements During West German Summertime Conditions in Support of Grafenwoehr Tests," ASL-TR-0066, September 1980.
138. Vechione, W. J., "Evaluation of the Environmental Instruments, Incorporated Series 200 Dual Component Wind Set," ASL-TR-0067, September 1980.
139. Bruce, C. W., Y. P. Yee, B. D. Hinds, R. G. Pinnick, R. J. Brewer, and J. Minjares, "Initial Field Measurements of Atmospheric Absorption at $9\mu\text{m}$ to $11\mu\text{m}$ Wavelengths," ASL-TR-0068, October 1980.
140. Heaps, M. G., R. O. Olsen, K. D. Baker, D. A. Burt, L. C. Howlett, L. L. Jensen, E. F. Pound, and G. D. Allred, "1979 Solar Eclipse: Part II Initial Results for Ionization Sources, Electron Density, and Minor Neutral Constituents," ASL-TR-0069, October 1980.
141. Low, Richard D. H., "One-Dimensional Cloud Microphysical Models for Central Europe and their Optical Properties," ASL-TR-0070, October 1980.

142. Duncan, Louis D., James D. Lindberg, and Radon B. Loveland, "An Empirical Model of the Vertical Structure of German Fogs," ASL-TR-0071, November 1980.
143. Duncan, Louis D., "EOSAEL 80, Volume I, Technical Documentation," ASL-TR-0072, January 1981.
144. Shirkey, R. C., and S. G. O'Brien, "EOSAEL 80, Volume II, Users Manual," ASL-TR-0073, January 1981.
145. Bruce, C. W., "Characterization of Aerosol Nonlinear Effects on a High-Power CO₂ Laser Beam," ASL-TR-0074, February 1981.
146. Duncan, Louis D., and James D. Lindberg, "Air Mass Considerations in Fog Optical Modeling," ASL-TR-0075, February 1981.
147. Kunkel, Kenneth E., "Evaluation of a Tethered Kite Anemometer," ASL-TR-0076, February 1981.
148. Kunkel, K. E., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF) White Sands Missile Range, New Mexico, August 1977 to October 1978, Part II, Optical Turbulence, Wind, Water Vapor Pressure, Temperature," ASL-TR-0077, February 1981.
149. Miers, Bruce T., "Weather Scenarios for Central Germany," ASL-TR-0078, February 1981.
150. Cogan, James L., "Sensitivity Analysis of a Mesoscale Moisture Model," ASL-TR-0079, March 1981.
151. Brewer, R. J., C. W. Bruce, and J. L. Mater, "Optoacoustic Spectroscopy of C₂H₆ at the 9μm and 10μm C¹²O₂¹⁶ Laser Wavelengths," ASL-TR-0080, March 1981.
152. Swingle, Donald M., "Reducible Errors in the Artillery Sound Ranging Solution, Part I: The Curvature Correction" (U), SECRET, ASL-TR-0081, April 1981.
153. Miller, Walter B., "The Existence and Implications of a Fundamental System of Linear Equations in Sound Ranging" (U), SECRET, ASL-TR-0082, April 1981.
154. Bruce, Dorothy, Charles W. Bruce, and Young Paul Yee, "Experimentally Determined Relationship Between Extinction and Liquid Water Content," ASL-TR-0083, April 1981.
155. Seagraves, Mary Ann, "Visible and Infrared Obscuration Effects of Ice Fog," ASL-TR-0084, May 1981.

156. Watkins, Wendell R., and Kenneth O. White, "Wedge Absorption Remote Sensor," ASL-TR-0085, May 1981.
157. Watkins, Wendell R., Kenneth O. White, and Laura J. Crow, "Turbulence Effects on Open Air Multipaths," ASL-TR-0086, May 1981.
158. Blanco, Abel J., "Extending Application of the Artillery Computer Meteorological Message," ASL-TR-0087, May 1981.
159. Heaps, M. G., D. W. Hoock, R. O. Olsen, B. F. Engebos, and R. Rubio, "High Frequency Position Location: An Assessment of Limitations and Potential Improvements," ASL-TR-0088, May 1981.
160. Watkins, Wendell R., and Kenneth O. White, "Laboratory Facility for Measurement of Hot Gaseous Plume Radiative Transfer," ASL-TR-0089, June 1981.
161. Heaps, M. G., "Dust Cloud Models: Sensitivity of Calculated Transmittances to Variations in Input Parameters," ASL-TR-0090, June 1981.
162. Seagraves, Mary Ann, "Some Optical Properties of Blowing Snow," ASL-TR-0091, June 1981.
163. Kobayashi, Herbert K., "Effect of Hail, Snow, and Melting Hydrometeors on Millimeter Radio Waves," ASL-TR-0092, July 1981.
164. Cogan, James L., "Techniques for the Computation of Wind, Ceiling, and Extinction Coefficient Using Currently Acquired RPV Data," ASL-TR-0093, July 1981.
165. Miller, Walter B., and Bernard F. Engebos, "On the Possibility of Improved Estimates for Effective Wind and Temperature," (U), SECRET, ASL-TR-0094, August 1981.
166. Heaps, Melvin G., "The Effect of Ionospheric Variability on the Accuracy of High Frequency Position Location," ASL-TR-0095, August 1981.
167. Sutherland, Robert A., Donald W. Hoock, and Richard B. Gomez, "An Objective Summary of US Army Electro-Optical Modeling and Field Testing in an Obscuring Environment," ASL-TR-0096, October 1981.
168. Pinnick, R. G., et al, "Backscatter and Extinction in Water Clouds," ASL-TR-0097, October 1981.
169. Cole, Henry P., and Melvin G. Heaps, "Properties of Dust as an Electron and Ion Attachment Site for Use in D Region Ion Chemistry," ASL-TR-0098, October 1981.

170. Spellacy, Robert L., Laura J. Crow, and Kenneth O. White, "Water Vapor Absorption Coefficients at HF Laser Wavelengths Part II: Development of the Measurement System and Measurements at Simulated Altitudes to 10 KM," ASL-TR-0099, November 1981.
171. Cohn, Stephen L., "Transport and Diffusion Solutions for Obscuration Using the XM-825 Smoke Munition," ASL-TR-0100, November 1981.
172. Pinnick, R. G., D. M. Garvey, and L. D. Duncan, "Calibration of Knollenberg FSSP Light-Scattering Counters for Measurement of Cloud Droplets," ASL-TR-0101, December 1981.
173. Cohn, Stephen L. and Ricardo Pena, "Munition Expenditure Model Verification: KWIK Phase I," ASL-TR-0102, December 1981.
174. Blackman, George R., "Cloud Geometry Analysis of the Smoke Week III Obscuration Trials," ASL-TR-0103, January 1982.
175. Sutherland, R. A., and D. W. Hoock, "An Improved Smoke Obscuration Model ACT II: Part 1 Theory," ASL-TR-0104, January 1982.
176. Lentz, W. J., "The Visioceilometer: A Portable Visibility and Cloud Ceiling Height Lidar," ASL-TR-0105, January 1982
177. Kennedy, Bruce W., "An Analysis of Craters Produced by Artillery Munitions During Dusty Infrared Test Series," ASL-TR-0106, January 1982.
178. Steinhoff, R. G., "Program Listings for EOSAEL 80-B and Ancillary Codes AGAUS and FLASH," ASL-TR-0107, February 1982.

**DATE
TIME**